Performance Analysis of Parallel Transmission and Multipath Routing in High-Speed Network Systems

Von der Fakultät für Elektrotechnik, Informationstechnik, Physik
der Technischen Universität Carolo-Wilhelmina zu Braunschweig

zur Erlangung des Grades einer Doktorin

der Ingenieurwissenschaften (Dr. -Ing.)

genehmigte Dissertation

von: Xiaomin Chen

aus: Jiangsu, China

eingereicht am: 21 März, 2014

mündliche Prüfung am: 19 Mai, 2014

Referent: Ashwin Gumaste, Professor, IIT Bombay
Referent: Dominic Schupke, Dr. -Ing, Airbus
Referent: Admela Jukan, Professor, TU Braunschweig
Vorsitzender: Thomas Kürner, Professor, TU Braunschweig
Druckjahr: 2014

Abstract

The unprecedented growth in Internet traffic is driving the steep upscaling of network capacity, which even in optical networks is expected to reach the so-called Shannon limit. The capacity challenge has thus led to innovations in all areas of networking. One of such innovations is the parallelization of end-systems combined with space division multiplexing in the network. As a result, parallel transmission combined with multipath routing is identified as a key solution to address the imminent capacity crunch and harvest the power of end-system hardware capabilities. The newly discovered benefits of Orthogonal Frequency Division Multiplexing (OFDM) networks are also driving the need to jointly design and analyze network system parallelism and multipath routing.

This thesis sets the goal to uniquely tackle the challenges associated with network capacity upscaling by unifying the research on multipath routing and parallel network system design. It starts by modeling multipath routing problems for data-intensive applications and thereafter identifies challenges and benefits of the same, while proposing novel and practically relevant solutions. Furthermore, the thesis presents pioneering studies in high-speed Ethernet parallel transmission in combination with various optical network technologies, from conventional WDM networks to advanced optical OFDM networks.

To this end, this thesis presents the first attempt to validate that high-speed Ethernet standard can benefit from the inherent parallelism of optical OFDM networks without resource penalty, such as spectrum fragmentation. Finally, this thesis addresses the critical issue of buffer dimensioning in high-speed parallel network systems, and outlines efficient solutions to address the same. Proposed novel

solutions in combination with linear network coding are shown to hold the key to an efficient high-speed Ethernet system design as they can significantly lower the buffer requirement at the receiver.

Kurzfassung

Das exponentielle Wachstum des Internetverkehrs treibt seit Jahren das steile Wachstum der Netzkapazitäten, wobei die Steigerung der Datenrate in optischen Netzen letztendlich durch das sogenannte Shannon-Limit begrenzt wird. Diese Herausforderung führt zu Innovationen in allen Bereichen der Netzwerktechnologie. Eine dieser Innovationen ist die Parallelisierung der Endsysteme in Kombination mit Multi-Path-Routing, welche die Aggregation der Leistung von Endsystemen erlaubt, allgemein auch als Antwort auf die Kapazitätskrise. Tatsächlich machen die kürzlich entdeckten Vorteile von optischen OFDM-Netzen ein gemeinsames Design und Analyse von Netzwerkparallelität und Multi-Path-Routing notwendig.

Diese Dissertation beschäftigt sich primär mit den Problemen der Netzkapazitäts-Skalierung und vereint hierzu die Forschung aus den Bereichen Multi-Path-Routing und dem Design paralleler Netzwerksysteme. Zu diesem Zweck werden zuerst neue Multi-Path-Routing-Probleme aus dem Bereich der datenintensiven Anwendungen modelliert. Anschließend werden die daraus entstehenden Herausforderungen und möglichen Vorteile analysiert, die zu neuen und praktisch anwendbaren Lösungen führen. Außerdem werden in dieser Dissertation bahnbrechende Studien über parallele Datenübertragung über High-Speed Ethernet in Verbindung mit verschiedenen optischen Netztechnologien (von konventionellem WDM bis zu optischen OFDM) präsentiert.

Zu diesem Zweck wird hier der erste Versuch durchgeführt, um zu beweisen, dass das heutige High-Speed Ethernet von der inhärenten Parallelität des optischen OFDM profitieren kann, und zwar ohne die Fragmentierung des Spektrums. Zum Schluss befasst sich diese Dissertation mit dem komplexen Problem der Pufferdimensionierung in parallelen Hoch-geschwindigkeitsnetzen. Empfohlen werden neu entwickelte Methoden in Kombination mit linearem Networkcoding,

die ein neues Systemdesign für High-Speed Ethernet mit signifikant verringerten Anforderungen an die Puffergröße bieten.

Acknowledgments

My deepest gratitude goes first and foremost to Prof. Admela Jukan, my PhD advisor, for her invaluable guidance, remarkable supervision and efforts to provide an excellent atmosphere for research. I was very fortunate to have worked with Prof. Admela Jukan. She is not only a research advisor, but also a mentor. I have learned from her visionary thinking as a women scientist. Her wisdom, enthusiasm, knowledge and commitment to the highest standards have encouraged and motivated me during my study and will continue to inspire me in my later career. I would also like to thank Prof. Ashwin Gumaste and Dr. Dominic Schupke, who kindly accepted to be my examiners, for their valuable time to evaluate my work. I am also very grateful to Prof. Thomas Kürner who generously accepted to chair the examination committee.

I was extremely fortunate to have some wonderful people as colleagues. A special thanks to Dr. Mohit Chamania, Marcel Caria, Marek Drogon, Dr. Tamal Das, Dr. Silvana Greco and Dr. Said Zaghloul, for being wonderful colleagues and great friends. Many thanks to Dr. Wolfgang Bziuk, who has been a great researcher and supportive colleague. Special thanks also goes to our administrative staff, Ms. Ruth Kiepert, Ms. Christiane Geller and Ms. Nadine Becker for their support during the course of my study. Especially I would like to thank Ms. Ruth Kiepert for her valuable time, patience to help me with my German. A very special thanks goes to my friends, Li Li and Gang Zhou, who have always been so great and stood by me through many difficulties in Germany.

Special thanks also go to my former professors at Tongji University for the remarkable learning experience during the course of my Bachelor and Master studies. Words will always be inadequate to thank my family for their constant support and encouragement. I am also very grateful to a lot more who have helped me and encouraged me and were not mention here, thank you very much.

List of Figures

List of Tables

Contents

1

Introduction

The advances in computing, networking, and storage technologies have ushered in unprecedented opportunities for parallel and distributed computing applications, including social media, big data, high-definition video streaming, cloud computing as well as large-scale scientific and engineering applications. The growing demands from these applications have forced network operators and carriers to cope with ever-increasing demand for bandwidth, which in turn has driven cross-board innovations in networking. Innovations in the field of optical network have focused on development of affordable high-bandwidth transmission systems and efficient utilization of optical spectral resource.

As the dominant optical switching and transmission technology of the last decade, Dense Wavelength Division Multiplexing (DWDM) has been abundantly deployed and developed since the mid 1990s. Today, the coarse granularity of DWDM channels, typically in "fixed-grid" spectral distances of 50GHz or 100GHz, is showing its fundamental capacity limits. To accommodate a massive Internet traffic growth at an annual rate of more than 30%, new research and development have started towards highly flexible and scalable network technologies beyond DWDM, most notably in the area of *elastic optical networks*, where connections can be provided with *just enough*

spectrum.

Despite the greatly improved spectral efficiency of elastic optical networks, at the current rate of traffic growth, optical networks are still expected to reach the so-called *optical capacity crunch* [8]. Studies have shown that capacity upgrades possible with the current technologies of single mode fiber systems have slowed down from about 80% per year to about 20% per year since 2002 [8]; it is expected that the capacity will eventually reach the Shannon limit in the near future. One way to address this capacity crunch is to utilize multi-level modulation formats to increase transmission rates per wavelength. However, increase bit rates in serial interface is at the expense of reduced all-optical transmission distance, i.e., higher the bit rate, shorter the optical reach.

Hence, the only option left to address the imminent Shannon limit on channel capacity is so-called *parallel transmission* [8], which combines the parallelization of end-systems with space division multiplexing in the networks. In a parallel transmission system, incoming serial traffic is first parallelized to multiple parallel flows or streams, and thereafter spatially distributed over multiple paths in the networks. As such, parallel transmission can harvest the capacity of networks and power of end-system capacities simultaneously.

In networks, space division multiplexing has been studied under the term *multipath routing*, which transmits traffic along multiple paths from source to destination. Up to now, multipath routing has not been widely adopted, due to the complexity issues, e.g., protocols and algorithms. It has been primarily used for load balancing in packet-switched networks, where the aggregated flows are distributed in a flow-based fashion [9]. However, using multipath for a large unicast flow, which is typically seen in data-intensive applications, is more challenging. To ensure correct reception of original flow, care has to be taken for traffic splitting at the source, modification of routing protocols in the network and realignment at the

destination.

In end-systems, parallelization is driven by economic and technological factors. On one hand, network device manufacturers and service providers are highly cost sensitive. Parallel transmission provides the immediate high-volume availability with cheaper components, which also enables to take full advantage of existing fiber cables in optical networks. On the other hand, the development of electrical interfaces is not necessarily coupled with the increase of wavelength capacity in fibers. As a result, high-speed end-systems, such as high-speed Ethernet, resorts to utilize parallel optics. For instance, 40Gbps Ethernet utilizes four fibers for each direction, i.e., transmitting and receiving, respectively [2].

To support the parallelization in the Ethernet layer, ITU-T has extended the Optical Transport Network (OTN) information structure. The new data containers are defined, including OTU2e for 10 Gbps Ethernet, OTU3e2 for 40 Gbps Ethernet and OTU4 for 100 Gbps Ethernet. The mature control functionality of OTN layer can be fully used to enable parallel transmission in optical networks. On the other hand, the parallelism in end-systems of Ethernet layer enables to utilize parallel low-speed optical paths in the optical layer [10].

We envision that the prospect of multipath routing in optical networks in combination of parallelization in high-speed end-systems holds the key to future networks and thus raises a number of exciting new challenges. The challenges range from system design to practical implementation. Solutions have to be economic, backward-compatible with existing optical infrastructure as well as visionary and far-reaching. A large portion of challenges is also architectural and algorithmic in nature. Given the fact that client traffic is spatially multiplexed and parallelized onto multiple paths, a common challenge to be addressed is *differential delay issue*, which results in traffic arriving at the destination in an incorrect order. In an optical system, two types of differential delay exist, viz., fiber effects-caused

differential delay, and path diversity-caused differential delay. Thus, the management of differential delay here requires consideration of fiber propagation properties. Also, to compensate differential delay in parallel transmission, buffer is required. However, buffering at speed of multiple 10 Gbps is challenging and costly. An affordable buffer dimensioning is thus vitally important to facilitate a viable deployment of parallel transmission systems.

1.1 Thesis Contributions

This thesis contributes to addressing challenges in modeling and system aspects pertaining to practical implementation of multipath routing and high-speed Ethernet parallel transmission over optical networks. The thesis models a multipath routing problem for the case of splitting large unicast flows and proposes solutions to the challenges and issues that hinder practical deployment of multipath routing in single- and multi-domain networks. It also presents the protocol extensions and implementation of multipath routing in Path Computation Element (PCE), which is a standardized network element for path computation as specified in IETF standards [11] [12].

For the parallelized end-systems, e.g., high-speed Ethernet, primary contributions focus on modeling and design of novel architectures that would provide a practical view on how parallel transmission in optical networks can be implemented with consideration of specific end-system features. The thesis also presents novel approaches for buffer dimensioning and reduction, thus addresses the inter-lane skew issue in high-speed Ethernet parallel transmission or differential delay issue in general in multipath routed systems. We propose to use linear network coding in parallel transmission to lower the requirements on optimality of paths used in parallel transmission and reduce buffering at the receiver. The solutions presented in this thesis are designed for practical deployment, and therefore

build upon existing standards.

The specific contributions of the thesis are summarized as follows:

1.1.1 Design, Modeling and Implementation of Multipath Routing for Data-intensive Applications

We present solutions to address the challenges and issues in multipath routing in optical networks from all perspectives, including novel algorithm design, protocol extension and implementation. We formulate a new multipath routing problem for large unicast flows. We address the challenges and issues of multipath routing in single-domain networks, and propose solutions for multi-domain service provisioning. We propose two novel multi-domain multipath computation schemes in full compliance with PCE standards [11] [13] [14], namely, *segmental multipath computation* and *end-to-end multipath computation*.

We design heuristic algorithms tailored for two representative data-intensive applications: bulk data transfer and real-time streaming. Finally, we complete the study with protocol extensions and implementation of multipath routing using an open-source PCE [15] [16]. The multipath routing problem studied in this thesis is different from the well investigated multipath routing problem that focuses on flow-based traffic distribution. It is formulated for large unicast flows from data-intensive applications, thus more challenging.

1.1.2 System Design and Modeling of High-speed Ethernet Parallel Transmission over Optical Networks

The novel systems proposed in this thesis lay foundations for practical implementation of parallel transmission in optical networks deploying different technologies, ranging from conventional *fix-grid* optical networks (WDM) to recently emerged *flexi-grid* (OFDM-based) optical networks.

We propose the first architectures for high-speed Ethernet parallel transmission over OTN/WDM and optical OFDM networks. The proposed architectures are fully compatible with high-speed Ethernet standardized in IEEE 802.3ba [2] and OTN specified in ITU-T G.709 [17]. The proposed architectures include all the relevant design parameters including data mapping between Ethernet and optical layer, buffer availability and differential delay as well as modulation formats. It is the first work to propose a backward-compatible solution for high-bandwidth applications. Especially, we are the first to propose that high-speed Ethernet can benefit from the inherent parallelism of optical OFDM networks without resource allocation penalty, such as the spectrum fragmentation.

We propose optimization models for parallel transmission in different networks: Multipath Routing and Wavelength Assignment (MRWA) in OTN/WDM networks and Multipath Routing and Spectrum Allocation (MRSA) in OFDM based elastic optical networks. To date, most of the RWA and RSA algorithms have focused on finding a single path solution. Our work is one of the first attempt to apply multipath routing in WDM and elastic optical networks. We also present a mechanism for optical parallel transmission without buffering, referred to as *bufferless parallel transmission*. In elastic optical networks, spectrum fragmentation is one of the prominent issues, especially in case of high-bandwidth applications. We design our algorithms with special focus on minimizing the spectrum fragments resulted from Routing and Spectrum Allocation (RSA). We show that parallel transmission in optical OFDM networks can effectively reduce spectrum fragmentation and improve the spectral efficiency. In addition to the optimization models, we also propose various heuristic algorithms and evaluate their effectiveness.

1.1.3 Novel Approaches to Facilitate Practical Deployment of High-speed Ethernet

Considering the buffer issue in the practical implementation of parallel transmission, we propose to use linear network coding in combination with parallel transmission. We design a novel high-speed Ethernet transmission system which applies network coding in end-systems at the price of small coding overhead, to consequently achieve efficient traffic management and reduce buffering of multipath routing. The parallel transmission system with linear network coding proposed in this thesis can lower the optimality requirements of multipath routing and reduce the buffer size at the destination. Our work carries potential to facilitate deployment of high-speed Ethernet, since it addresses the major obstacle in parallel transmission.

We present a theoretical framework for parallel transmission systems with linear network coding and analyze the coding overhead. We also propose a novel buffer model for parallel transmission system with linear network coding. Based on the proposed buffer model, we derive an upper bound of the buffer size required for decoding, and show that it is still smaller than the buffer size required by re-ordering in conventional systems without linear network coding. The reduction of buffer required is crucial for practical deployment of high-speed Ethernet, where buffering and re-ordering present the major challenge.

1.2 Supporting Publications

1.2.1 Book Chapter

1. FIA2011, Pascale Vicat-Blanc, Sergi Figuerola, Xiaomin Chen, Giada Landi, et.al., “Bringing Optical Networks to The Cloud: An Architecture for A Sustainable Future Internet”, SpringerLink, 2011

1.2.2 Journal Articles

1. X. Chen, A. Engelmann, A. Jukan, M. Médard, "Linear Network Coding Reduces Buffering in High-Speed Ethernet Parallel Transmission Systems," IEEE Communication Letters, Volumn 8, Issue 4, April 2014, PP. 636-639.

2. X. Chen, A. Jukan, A. Gumaste, "Optimized Parallel Transmission in Elastic Optical Networks to Support High-Speed Ethernet," IEEE/OSA Journal of Lightwave Technology (JLT), Volume 32, Issue 2, January 2014, PP. 228-238.

3. X. Chen, A. Jukan, "Optimized Parallel Transmission in OTN/ WDM Networks to Support High-Speed Ethernet with Multiple Lane Distribution (MLD)", IEEE/OSA Journal of Optical Communications and Networking (JOCN), Volume 4, Issue 3, March 2012, PP. 248-258.

4. X. Chen, A. Drummond, N. da Fonseca, A. Jukan, "Multipath Routing with Topology Aggregation for Scalable Inter-domain Service Provisioning in Optical Networks", Elsevier Journal of Optical Switching and Networking (OSN), Volume 9, Issue 4, November 2012, PP. 314-322.

5. M. Chamania, X. Chen, A. Jukan, F. Rambach, M. Hoffmann, "An Adaptive Inter-domain PCE framework to Improve Resource Utilization and Reduce Inter-domain Signaling," Elsevier Journal of Optical Switching and Networking, Volume 6, Issue 4, December 2009, PP. 259-267.

1.2.3 Conference and Workshop Papers

1. X. Chen, A. Jukan, M. Médard, "A Novel Network Coded Parallel Transmission Framework for High-Speed Ethernet," in

IEEE Global Telecommunications Conference (GLOBECOM), December 2013.

2. X. Chen, Y. Zhong, A. Jukan, "Multipath Routing in Path Computation Element (PCE): Protocol Extensions and Implementation," in 18th European Conference on Network and Optical Communications (NOC), July 2013.

3. X. Chen, Y. Zhong, A. Jukan, "Multipath Routing in Elastic Optical Networks with Distance-adaptive Modulation Formats,"in IEEE International Conference on Communications (ICC), June 2013.

4. X. Chen, A. Jukan, A. Gumaste, " Multipath De-fragmentation: Achieving Better Spectral Efficiency in Elastic Optical Path Networks," in IEEE International Conference on Computer Communications (INFOCOM), April 2013, PP. 390-394.

5. X. Chen, M. Chamania, A. Jukan, "On the Effectiveness of Optical Parallel Transmission in IP Offloading," in 18th European Conference on Network and Optical Communications (NOC), June 2012.

6. X. Chen, A. Jukan, A. Gumaste, "On the Usage of FDLs in Optical Parallel Transmission to Support High Speed Ethernet," in International Conference on Optical Network Design and Modeling (ONDM), April 2012.

7. J. Santi. A. Drummond, N. da Fonseca, X. Chen, A. Jukan, "Leveraging Multipath Routing and Traffic Grooming for an Efficient Load Balancing in Optical Networks," in IEEE International Conference on Communications (ICC), June 2012.

8. X. Chen, M.Chamania, A. Jukan, "An Evolutionary Framework to Facilitate the Inter-domain Multipath Routing in Car-

rier Networks," in IEEE Conference on Computer Communications (INFOCOM) Hight-speed Networks Workshop, April 2011.

9. X. Chen, A. Jukan, A. C. Drummond, N. da Fonseca, "A Multipath Routing Mechanism in Optical Networks with Extremely High Bandwidth Requests," in IEEE Global Telecommunications Conference (GLOBECOM), December 2009.

10. X. Chen, M. Chamania, A. Jukan, A. Drummond, N. da Fonseca, "On the Benefits of Multipath Routing for Distributed Data-intensive Applications with High Bandwidth Requirements and Multidomain Reach," in Communication Networks and Services Research Conference (CNSR), May 2009, PP. 110-117.

11. X. Chen, M.Chamania, A. Jukan, A. C. Drummond, N. da Fonseca, "QoS-Constrained Multi-path Routing for High-End Network Applications", in IEEE Conference on Computer Communications (INFOCOM) Hight-speed Networks Workshop, April 2009.

12. M. Chamania, X. Chen, A. Jukan, F. Rambach, C. Gruber, M. Hoffmann, "Embedding Optical Ethernet Services within the Path Computation Element Framework: The 100GET Approach," in Optical Fiber communication/National Fiber Optic Engineers Conference (OFC/NFOEC), March 2009.

13. M. Chamania, X. Chen, A. Jukan, F. Rambach, C. Gruber, M. Hoffmann, "Adaptive Advance Reservation Based Inter-Domain Framework," in IEEE 2nd International Symposium on Advanced Networks and Telecommunication Systems (ANTS), December 2008 .

14. X. Chen, A. Jukan and T. Fischer, "End-to-End Service Provisioning in Carrier-Grade Ethernet Networks: The 100 GET-E3 Approach," in International Conference on Optical Network Design and Modeling (ONDM), March 2008.

15. A. Gumaste, A. Lodha, X. Chen, A. Jukan and N. Ghani, "A Novel Node Architecture for Light-trail Provisioning in Mesh WDM Metro Networks," Optical Fiber communication/National Fiber Optic Engineers Conference (OFC/NFOEC), February 2008.

1.3 Thesis Organization

This thesis is structured as follows. After the introduction, Chapter 2 models and evaluates multipath routing for data-intensive applications and presents the relevant protocol extensions and implementations. Chapter 3 first presents the transition from multipath routing to parallel transmission driven by the parallelization in the end-systems of high-speed Ethernet. Afterwards, it models and evaluates parallel transmission in OTN/WDM networks. Chapter 4 is dedicated to the modeling of parallel transmission in elastic optical networks, including uniform and distance-adaptive modulation formats assignment. It proposes an architecture which shows how high-speed Ethernet transmission can be supported in OFDM-based optical networks. It includes both Integer Linear Programming (ILP) based optimization models and heuristic algorithms. Chapter 5 proposes a novel system model which applies network coding in end-systems and at the price of small coding overhead, to consequently reduce buffering. This chapter also presents analytical derivation of an upper bound of the buffer size required for decoding. Finally, this chapter presents a case study on network parallel transmission in optical OFDM networks. Chapter 6 concludes the thesis and provides the directions for future work.

2

Multipath Routing in Optical Networks for Data-intensive Applications

2.1 Introduction

The past decade has witnessed significant growth in commercial and scientific applications, such as high-resolution scientific visualization, high-quality, real-time consumer-driven media production and distribution, big data transfer, and to name a few [18]. Such applications are generally data-intensive and dependent on widely distributed supercomputing and data-storage facilities, hence pose new challenges to network infrastructures, service provisioning paradigms and technologies. The optical transport networks are required to be capable of not only providing sufficient bandwidth, but also being highly adaptive to widely fluctuating bandwidth demands of diverse applications.

Traditionally, single path routing is performed in optical networks, which is sufficient for most connection requests. However, it may be difficult to find a single path for a data-intensive application transferring *tera*- and *pera*-scale data. Simply upgrading network capacity for data-intensive applications is not economic. Moreover, link capacity, even in a single mode fiber, is expected to reach Shannon limit in the near future [8], whereas Internet traffic continues to boom. It

is clear that a novel network service paradigm, which can support high-bandwidth applications, is imperative for the next generation networks.

Multipath routing is a valid solution to address this challenge [19] [20]. It computes multiple paths between source and destination. By aggregating available bandwidth from multiple paths, large flows from a data-intensive application, such as high-definition real-time media streaming or big data transfer, can be supported in a bandwidth limited network. In fact, multipath routing has been standardized and widely used in optical networks, known as *inverse-multiplexing*. A representative technique is Virtual Concatenation (VCAT) which defines logical channels with different line rates. A large capacity payload is divided into multiple smaller capacity payload containers and transmitted over logical channels. To date, VCAT has been specified and standardized for Synchronous Optical Networking (SONET)/Synchronous Digital Hierarchy (SDH) [21][22] and Optical Transport Network (OTN) networks [17].

Despite of all the advantages, multipath routing has not been widely adopted so far, primarily due to the associated complexity. Care has to be taken in multipath routing in terms of traffic splitting and reassembly. Discrepancy exists among paths, such as available bandwidth and propagation delay, which can impact on performance of the connection. For instance, a real-time streaming application is very sensitive to jitter. When it is not possible to set up a connection along a single path, multipath routing needs to be designed such that the packets/frames arrive at the receiver within a tolerable time. Commonly, buffering is required at the receiver in case of multipath routing to compensate the difference among transmission delays on different paths. Cost and availability of high-speed memory devices also hinder the practical implementation of multipath routing. As a result, multipath routing resorts to flow-based distribution, such as the well known Equal Cost Multi-Path routing (ECMP) protocol.

Each flow is assigned to a single path to ensure in-order delivery. Finally, more and more applications are dependent on computing and storage resources in the cloud. Nowadays, the cloud resources are globally distributed, which may lead to a connection across multiple administrative domains. Multipath routing in combination with inter-domain service provisioning is rather challenging, and has not be addressed so far.

This chapter presents solutions to address the challenges and issues of multipath routing from all perspectives, including novel algorithm design, relevant protocol extension and implementation; and proposes novel service provisioning scheme for multi-domain scenarios. We first formulate a multipath routing problem as a profit maximization problem, where the revenue is the total achievable bandwidth and the cost is the buffer to compensate differential delay. We propose to apply multipath routing for a scalable inter-domain service provisioning and present two novel multi-domain multipath computation schemes in full compliance with PCE standards [11] [13] [14], namely, *segmental multipath computation* and *end-to-end multipath computation*. We design heuristic algorithms tailored for two representative data-intensive applications: bulk data transfer and real-time streaming. Finally, we complete the study with protocol extensions and implementation based on an open-source PCE [15] [16].

2.2 Supporting Publications

1. X. Chen, Y. Zhong, A. Jukan, "Multipath Routing in Path Computation Element (PCE): Protocol Extensions and Implementation," in 18th European Conference on Network and Optical Communications (NOC), July 2013.

2. X. Chen, A. Drummond, N. da Fonseca, A. Jukan, "Multipath Routing with Topology Aggregation for Scalable Inter-domain

Service Provisioning in Optical Networks," Elsevier Journal of Optical Switching and Networking (OSN), Volume 9, Issue 4, November 2012, PP. 314-322.

3. X. Chen, M.Chamania, A. Jukan, "An Evolutionary Framework to Facilitate the Inter-domain Multipath Routing in Carrier Networks," in IEEE Conference on Computer Communications (INFOCOM) Hight-speed Networks Workshop, April 2011.

4. X. Chen, A. Jukan, A. C. Drummond, N. da Fonseca, "A Multipath Routing Mechanism in Optical Networks with Extremely High Bandwidth Requests," in IEEE Global Telecommunications Conference (GLOBECOM), December 2009.

5. X. Chen, M. Chamania, A. Jukan, A. Drummond, N. da Fonseca, "On the Benefits of Multipath Routing for Distributed Data-intensive Applications with High Bandwidth Requirements and Multidomain Reach," in Communication Networks and Services Research Conference (CNSR), May 2009, PP. 110-117.

6. X. Chen, M.Chamania, A. Jukan, A. C. Drummond, N. da Fonseca, "QoS-Constrained Multi-path Routing for High-End Network Applications," in IEEE Conference on Computer Communications (INFOCOM) Hight-speed Networks Workshop, April 2009.

2.3 Benefits and Issues of Multipath Routing

A typical application scenario of multipath routing is shown in Fig. 2.1, where large scale data transfer is required between two data centers. Such applications commonly require to finish data transmission within a given time frame, and data volume can be Terabits or even

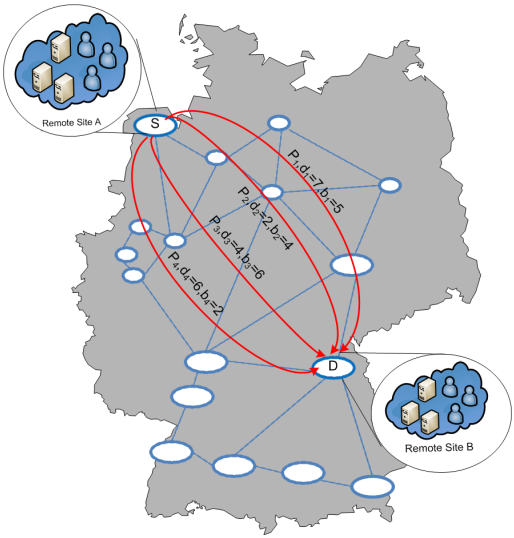


Figure 2.1: Large-scale data transfer between two distributed data centers

Perabits. The bandwidth requirement may be beyond the capacity of the interconnecting networks. Assume that 1 Tb data is required to be transmitted from site A to site B within 30s and the transport network is a Wavelength Division Multiplexing (WDM) network in which capacity per wavelength is 10 Gbps. It is clear that at least four wavelengths are required between two sites.

Although it is expected that data-intensive applications can benefit from multipath routing, numerous challenges and issues arise, especially for those with certain Quality of Service (QoS) requirements. For instance, multiple paths may present differences in the end-to-end delay. It can cause jitter at the destination, which is a critical QoS requirement of real-time streaming applications. The phenomena caused by the discrepancy among paths is commonly re-

ferred to as *differential delay* [19] and requires buffering of packets from all other paths till the packets from the path with the highest delay arrive. Hence, a multipath routing algorithm requires a careful design which can achieve a trade-off between achievable bandwidth and differential delay.

The memory size required at the destination depends on the chosen paths and their available bandwidth. The former decides the differential delay, while the latter determines traffic splitting ratio, i.e., how much traffic can be distributed to a chosen path. To illustrate the impact of path selection and traffic splitting, we take the example shown in Fig. 2.1. Between source (S) and destination (D), there are four paths P_1, P_2, P_3, and P_4, each with a different available capacity b and path delay d. Requested is a dataset transmission from S to D with size of 40 units. With single shortest path first routing algorithms, P_2 is chosen, which results in the end-to-end transmission time of $40/b_2 + d_2 = 12$ units. In the case of data transmission over two paths, e.g., P_1 and P_3 are chosen to maximize the achievable bandwidth. The resulting transmission delay is then 11, calculated by $max.\{(20/b_1 + d_1), (20/b_3 + d_3)\}$. The size of memory required in this case is calculated by $(d_1 - d_3) \cdot b_3 = 18$ units. When minimization of differential delay is considered, P_1 and P_4 will be used as the optimal solution with a memory cost $(d_1 - d_4) \cdot b_4 = 2$ units. The end-to-end transmission time in this case is $max.\{20/b_1 + d_1, 20/b_4 + d_4\} = 16$ units. This example shows how different strategies in traffic splitting and path selection can lead to different connection setup, which makes multipath routing algorithm design rather challenging.

On the other hand, geographically distributed computing resources may cause data transmission cross multiple administrative domains. Multipath routing in a single domain is based on the assumption of full knowledge of the network states. In a multi-domain scenario, this is not realistic which makes multi-domain multipath routing

very challenging. Due to the constraints related to scalability, security and administrative policies, the intra-domain information cannot be fully advertised to other domains, which naturally leads to the limited view of the entire network. To date, it has not been addressed either.

2.4 Multipath Routing for Data-intensive Applications

In this section, we model the multipath routing for data-intensive applications and propose algorithms, respectively. To ease the understanding of problem formulation, we first introduce the used terminologies as follows:

- Link delay- time for a signal to traverse in a link; it is related to the signal speed and physical distance.
- Transmission delay - time it takes to push the packet's bits onto the link.
- Propagation delay - time for a signal to reach its destination.
- Path delay- accumulation of link delays along the path.
- Differential delay- time difference among delay of paths.

Given is a directed graph $G(V, E)$, where V is the set of nodes and E is the set of links. Each link $e \in E$ is associated with two parameters: link capacity c_e and link delay d_e. The delay of a path P is defined as:

$$d_P = \sum_{e \in P} d_e$$

The bandwidth of a path P is denoted by b_P

$$b_P = \min\{c_e\}, e \in P$$

The differential delay between two paths P and P' can be defined as:

$$dd(P, P') = |d_P - d_{P'}|$$

The memory size required at the destination is denoted as M_r and the maximum available memory size is assumed to be M_d. Assuming that the highest delay path in the solution is $\tilde{P}$, and given that the traffic on a path P is t_P, the total buffer size required is given by [23]:

$$M_r = \sum_{P \in \mathcal{P}} t_P \cdot (d_{\tilde{P}} - d_P) \tag{2.1}$$

Per Eq. (2.1), it is clear that the size of memory required for reordering depends on two factors:

1. Delays of chosen paths

2. Traffic on each path, i.e., traffic split ratio.

We therefore model the multipath routing problem as two subproblems, i.e., multipath computation and optimized traffic splitting.

2.4.1 Multipath Computation Algorithm

For tractability, we limit the number of paths that can be used for a connection to be bounded by N. It is a parameter decided by the network service provider based on the local policies, specific applications and path computation complexity etc. For instance, for jitter-sensitive applications, N should be as small as possible due to the increased variance in the chosen paths. On the other hand, large N is preferred by latency-sensitive applications such as bulk data transfer.

Algorithm 1: Multipath path computation algorithm

```
   Input: G(V, E), (s, d), N
 1 count = 0;
 2 Initialize an Empty Array of Paths pathSet;
 3 For all v_i ∈ V s.t. there exists a link e_j ∈ Efrom s to v_i, create a
   path from s to v_i and insert in pathSet;
 4 Sort pathSet in decreasing order of bandwidth;
 5 while count < N do
 6     count = 0;
 7     for (i = 0 to pathSet.length) do
 8         if (pathSet[i].destination == d) then
 9             Increase count by 1 ;
10             if (count == N) then
11                 break;
12             end
13         end
14         else
15             path = pathSet[i];
16             remove pathSet[i] from pathSet;
17             foreach (e_j ∈ E attached to path.destination) do
18                 Let the vertices at the end of e_j be v_k and
                   path.endVertex;
19                 if (v_k ∉ path.vertices) then
20                     Create a new path by extending path with e_j;
                       Insert new path in pathSet using insertion sort;
21                 end
22             end
23         end
24     end
25     if (no more paths can be extended) then
26         break;
27     end
28 end
```

The proposed multipath computation algorithm (Alg. 1) computes the paths in decreasing order of available bandwidth [19]. The path computation begins at the source node, and creates path segments to all neighboring nodes.

An array named *pathSet* is initialized with all path segments beginning from the source to its neighboring nodes and sorted in the decreasing order of available bandwidth. In each iteration, we choose the first path in *pathSet* which does not terminate at the destination node, and remove it from *pathSet*. Then, for each possible extension of the chosen path, we check if a loop is formed in the path segment. This is done by checking if the new vertex that the path segment is extended to exists in the path segment. If the new path segment is valid, it is inserted in *pathSet*. When the selected path segment is terminated at the destination, the next path segment is chosen.

In order to calculate the first N widest shortest paths, the sorting algorithm check the delay of the paths in question in case of a tie. At the beginning of each iteration, we check the number of consecutive paths at the beginning of *pathSet* terminating at the destination node. If N paths have been computed, the algorithm is terminated and the computed paths are selected.

2.4.2 Optimization Model for Path Selection and Traffic Splitting

We propose an ILP based optimization model to select optimal paths from the path set computed by Alg. 1 and determine the optimal traffic splitting ratio among the chosen paths.

Assume a large flow is distributed to the chosen paths, each path is then assigned with a fraction of the original flow. The amount of traffic that can be distributed to a path P is bounded by the available bandwidth for the current connection request. We hereby define two cost factors for each individual flow on path P and the cost caused by the memory at the receiver, denoted as C_F and C_M, respectively. The relative scale of the two factors defines whether minimization of differential delay or maximization of flow plays a dominant role. For example, in case of a large C_F, the memory cost may become insignificant, and the problem reduces to a flow maximization problem, while in case where the memory cost is significantly high, the

cost of memory may result in solutions where only the widest path is chosen.

The proposed ILP optimization model relies on the following variables:

- $\mathbf{x_e}$ - an integer variable denoting the traffic on link e
- $\mathbf{t_P}$ - an integer variable denoting the flow distributed to path P
- $\tilde{P}$ - the path with the highest delay among all chosen paths

$$\textbf{Objective} : \text{Maximize} \sum_{P \in (P)} C_F \cdot \mathbf{t_P} - C_M \cdot M_r$$

Subject to:

$$\forall e \in E : \ \mathbf{x_e} = \sum_{P \in \mathcal{P} \wedge e \in P} \mathbf{t_P} \tag{2.2}$$

$$\forall e \in E : \ \mathbf{x_e} \leq c_e \tag{2.3}$$

$$M_r = \sum_{P \in \mathcal{P}} \mathbf{t_P}(d_{\tilde{P}} - d_P) \tag{2.4}$$

$$\forall P \in \mathcal{P} : \mathbf{t_P} = 0 \, if \, d_P > d_{\tilde{P}} \tag{2.5}$$

$$\forall P \in \mathcal{P} : \mathbf{t_P} >= 0 \, if \, d_P <= d_{\tilde{P}} \tag{2.6}$$

$$\mathbf{t_{\tilde{P}}} > 0 \tag{2.7}$$

$$\sum_{P \in (P)} \mathbf{t_P} >= B_{min} \tag{2.8}$$

$$M_r \leq M_d \tag{2.9}$$

The variable x_e is defined as the sum of flows in the different paths that pass through the link $e \in \mathcal{E}$, as shown in Eq. (2.2). The constraint in Eq. (2.3) ensures that the total traffic on a link does not exceed the total capacity of the link itself. Eq. (2.4) calculates the memory size at the destination node, with the assumption that $\tilde{P}$ is the highest delay path in the solution. Eq. (2.5) and Eq. (2.6)

are constraints to ensure all flows are positive, and no path with delay greater than $d_{\tilde{P}}$ is selected. Eq. (2.7) ensures that $\tilde{P}$ is chosen by ensuring that the flow $t_{\tilde{P}}$ is not zero. Eq. (2.8) defines that the total bandwidth can not be less than a pre-defined value B_{min}. Eq. (2.9) indicates that the required memory size can not exceed the maximum memory boundary in the network. The value of B_{min} is derived from the total size of the data to be transferred F and the maximum delay constraint D_{max}. If the file size is sufficiently large, the link delay can be assumed to be negligible as compared to the time taken to send data to a link. The maximum delay constraint is:

$$\sum_{P \in (P)} \mathbf{t_P} >= \frac{F}{D_{max}} \tag{2.10}$$

It can be seen that the requested memory size in Eq. (2.4) depends on the choice of the highest delay path. As the calculation of the memory depends on the solution, to solve the ILP , we use a certain $P \in \mathcal{P}$ as a potential value of $\tilde{P}$. Each choice of $\tilde{P}$ implies that certain paths may not be taken into consideration as their delay is greater than the delay of $\tilde{P}$. Therefore, the ILP is run $N = |\mathcal{P}|$ times, once for each possible candidate for $\tilde{P}$, and the solution with the highest objective among the N iterations is the optimal solution.

The time complexity of an ILP is known to be exponential. In the worst case scenario, the ILP would be run N times, and the i-th iteration has an input path set of size i. Therefore the time complexity of the ILP is in the order of $O(2^1 + 2^2 + .. + 2^N)$ which is equal to $O(2^{N+1})$.

2.4.3 Evaluation

We evaluate the proposed optimization model against various network parameters such as network load, memory cost. Germany 17 network topology [1] shown in Fig. 2.2 is used in the performance

evaluation. We simulate a dynamic network, with connection arrivals following a Poisson process with inter-arrival times following a negative exponential distribution. Each connection requests a bandwidth of 1 Gbps between a random pair of nodes. Please note that all the dynamic connection requests are used to generate network load. At each network load, we generate a request for multipath routing and run Alg. 1. We aim to show how much bandwidth can be aggregated by using proposed algorithm. The bandwidth of the widest shortest path, which is used otherwise by single path routing, is shown as a comparison.

For each path set computed by Alg.1, we run the ILP optimization model with different C_F and C_M values and show the impact of cost factors in the path selection and traffic splitting. Finally, we study the relation between the memory cost (C_M) and achievable bandwidth in multipath routing. The link capacity is assumed to be 40 Gbps and the delay of each link is estimated on the basis of the geographical distance between the two nodes. We set the unit of C_F to be Gbps^{-1} and memory cost has the unit of ms/Gbps. The multipath computation algorithm is evaluated with an event-driven simulator implemented in Java. The ILP optimization model is implemented in Gurobi Optimizer [24].

Fig. 2.3 shows that increase of network load leads to the decrease of the achievable bandwidth of both single and multipath routing. Also, increase in memory cost implies that fewer paths are chosen due to the increase of differential delay. To show the relation between memory cost and achievable bandwidth, we observe the change of achievable bandwidth at 130 Erlang. As shown in Fig. 2.4, the total achievable bandwidth of the optimal solution decreases rapidly with increase of memory cost. From this study, it is clear that memory cost is a major obstacle in practical deployment of multipath routing.

Figure 2.2: Germany 17 network topology [1]

2.5 Multi-domain Multipath Computation Schemes

As discussed in the previous section, geographically distributed computing resources and storages lead to the necessity of inter-domain service provisioning. However, a domain only exposes limited internal information to other domains due to administrative or technical reasons. The limited visibility makes multi-domain multipath computation rather challenging.

Path Computation Element (PCE) is a system component, application, or network node that is capable of computing a path between a source and a destination [11]. The PCE systems work in a server-client fashion, i.e., the PCE server computes a path upon the request from a Path Computation Client (PCC) based on the information maintained in its Traffic Engineering Database (TED). The com-

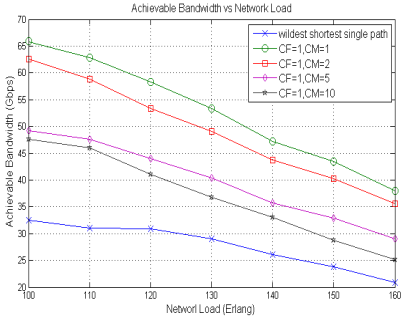


Figure 2.3: Bandwidth vs. network load

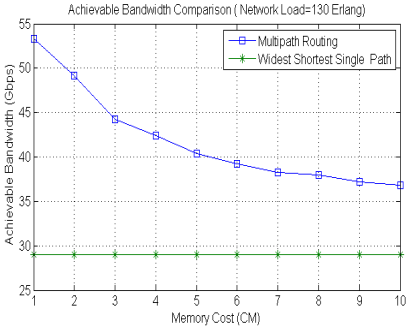


Figure 2.4: Achievable bandwidth vs. memory cost at 130 Erlang

munication between the PCE server and its clients is based on PCE communication protocol (PCEP) [12]. PCE has become the *de-facto* standard for path computation with QoS requirements, such as bandwidth, end-to-end delay etc [11]. To date, however, PCE-based service provisioning schemes have mainly focused on single path routing only.

In this section, we propose, for the first time, PCE-based multi-

domain multi-path computation schemes [20], referred to as *segmental multipath computation* and *end-to-end multipath computation*, respectively.

2.5.1 Segmental Multipath Computation

In inter-domain service provisioning, it is possible that some transit domains can not support the connection requests, due to insufficient network resource. With single path routing, the path computation procedure has to *crankback* [13] to the previous domain and find an alternative domain as the next hop. Single path routing with crankback mechanism has relatively high signaling overhead and takes long time to find find an end-to-end path.

In contrary, multipath routing can be applied to address this issue. Instead of crankback to another domain, multipath routing can be used in the heavily loaded transit domain and convey traffic to the next domain. Upon receiving connection request, domain PCE carries out path computation based on either single path routing algorithm or multiple path routing algorithm, depending on the available network resource. The information of the computed path is sent to PCE of the next domain. Destination domain makes the final decision on the end-to-end path(s) based on all information re-

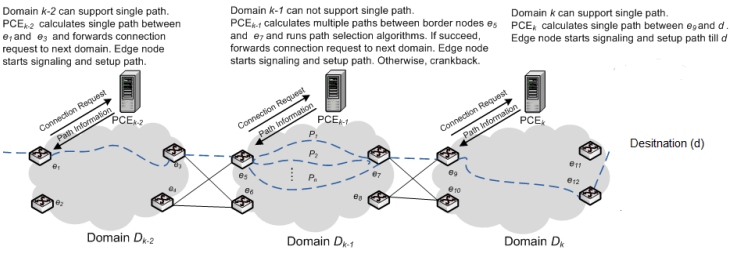


Figure 2.5: Segmental multipath computation scheme

ceived from previous domains. Given the fact that the final solution is composed of paths computed from all transit domains, we refer it to as *segmental multipath computation*.

An example is shown in Fig. 2.5. PCE_{k-2} succeeds to compute a single path for the request between border nodes e_1 and e_3. The index k denotes the order of domains along the domain chain, where $k = 1$ is the source domain. Path information is sent back to e_1 and the request is forwarded to the next domain D_{k-1} which cannot find a single path the sufficient bandwidth for the connection request. In conventional single path routing with *crankback*, the connection request will be rejected by D_{k-1}. In segmental multipath computation scheme, PCE_{k-1} calculates multiple paths by running *k-shortest-path algorithm* [25] instead of sending *crankback* message to the previous domain. In Fig. 2.5, $\{P_1, P_2, .., P_n\}$ are calculated between border nodes e_5 and e_7 and PCE_{k-1} forwards the connection request to the next domain.

In segmental multipath computation scheme, each domain can make its own decision on the routing algorithms for the incoming requests. However, the major drawback of the *segmental multipath computation* scheme is that it cannot guarantee end-to-end delay since path setup is decided in a per-domain fashion.

2.5.2 End-to-End Multipath Computation

End-to-End Multipath Computation scheme can be applied in the scenarios where a virtual topology of the whole network can be constructed. Topology aggregation mechanisms, e.g., proposed in [26], can be used for topology information dissemination. Each domain advertises limited information about the domain topology, which is assimilated by other domains and used to construct an abstract inter-domain topology. Path computation schemes use the knowledge of these abstract schemes to determine actual paths in the network.

An example is shown in Fig. 2.6. Domains $\{D_1, D_2, .., D_k\}$ consti-

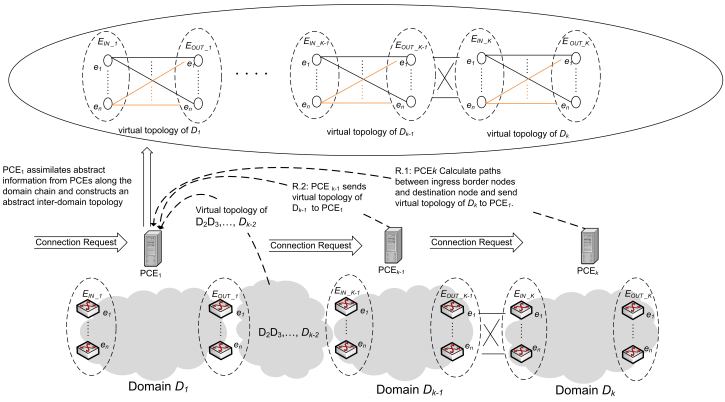


Figure 2.6: End-to-end multipath computation scheme

tute a domain chain between source and destination, which is known in advance. E_{IN_k} is the set of ingress border nodes of domain D_k while E_{OUT_k} is the set of egress border nodes. PCE$_k$ represents paths advertised for inter-domain traffic between E_{IN_k} and E_{OUT_k} in an aggregate topology. Upon receiving connection request, PCE$_2$ sends the aggregate topology of domain D_2 to PCE$_1$ and forward connection request to the next domain D$_3$. The same procedure is applied until the connection request arrives at D_k and the PCE$_1$ receives aggregate topology of the whole domain chain. PCE$_1$ in source domain runs multipath routing algorithms for incoming request with QoS requirements.

2.5.3 Topology Aggregation for Multi-domain Multipath Routing

As previously discussed, topology aggregation is a commonly used technique in multi-domain service provisioning, which abstracts domain topology and advertises it as a *virtual topology* to other domains. PCEs utilize the virtual topologies for inter-domain path

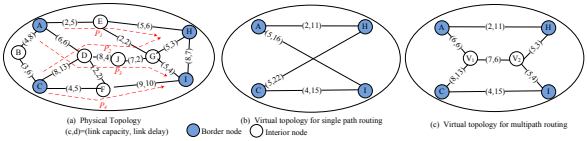


Figure 2.7: A topology aggregation example for inter-domain service provisioning with and without consideration of multipath routing

computation. So far, three representative virtual topologies have been proposed by Optical Internetworking Forum (OIF) for inter-domain service provisioning in optical networks. They are referred to as *Abstract Node*, *Abstract Link* and *Pseudo-node*, which represent *Single Node, Full mesh* and *Symmetric Star*, respectively. Multi-domain service provisioning based on virtual topologies was also discussed by ITU-T and OIF [27, 28], with the interface between the domains, defined as external network to network interface (E-NNI). To guarantee the resource for the inter-domain connection requests, transit tunnels can be set up with bandwidth reserved in advance between border nodes of a domain, which we refer to as *static virtual topology*.

However, existing topology aggregation methods are designed for single path routing, which do not indicate if a physical link is shared by multiple virtual links. This is illustrated in Fig. 2.7(a) where four virtual links are advertised with their available capacity and delay between border nodes. Conventional aggregate topology is represented in Fig. 2.7(b); this topology representation is commonly used in single path routing. However, path P_2 and P_3 share segment $D - J - G$ with available capacity as 7. When P_2 and P_3 are used simultaneously in multipath routing, the total capacity advertised as 10 violates the actual available capacity of shared segments.

We therefore propose an extension to the virtual topology representation, by representing shared segments in the virtual topology with their available capacity and delay accordingly, as shown

in Fig. 2.7(c). In the following section, we will present a case study which utilizes the proposed static virtual topology aggregation mechanism to facilitate multi-domain multipath routing.

2.6 Case Study: Multi-domain Multipath Routing for Data-intensive Applications

In this section, we propose algorithms tailored for two representative data-intensive applications, referred to as bulk data transfer and real-time streaming, respectively. The virtual topology composed of abstract domain topologies is denoted as $G(V, E)$, where V is the set of virtual nodes and E is the set of virtual links. We define a parameter β to constrain the percentage of available bandwidth on a path p to be allocated for a connection request. For instance, assuming the bandwidth required by a connection requests is B, and available bandwidth of p is B_p. Path p is sufficient only if $B_p \cdot \beta \geq B$. This parameter is defined to balance the network load and avoid congestions caused by high-bandwidth connections.

2.6.1 Algorithm for Bulk Data Transfer

Bulk data transfer is commonly seen in scientific computing, where data is frequently transferred from one site to another site for further processing. An example is the CyberShake research. Hundreds of terabytes of data need to be transferred between the research centers and TeraGrid center for earthquake forecast research [29]. The destination site is equipped with storage and the data processing starts after the data transmission finishes. Differential delay is not a critical issue for such applications. Instead, it is constrained by a given transmission deadline, that is, data needs to be transferred to the destination site within a fixed duration.

A connection request for the bulk data transfer applications is denoted as $R(s, d, C, D_R)$, where (s, d) are the source and destina-

tion, respectively. C is the size of the data chunk and D_R is the transmission deadline. The end-to-end delay of bulk data transfer is composed of path delay and processing delay at the source node. Assuming that data chunk size is C and the total bandwidth allocated to the connection is B, the time required to send out all data at the source node is calculated as C/B. To serve such applications, the end-to-end delay should not exceed the transmission deadline, i.e., $C/B + pd_{\tilde{p}} \leq D_R$, where $pd_{\tilde{p}}$ is the delay of the longest path used in the connection. The proposed algorithm for bulk data transfer is shown in Alg.2 [30].

The algorithm first calculates a candidate path set, referred to as $\mathcal{P}$, using K-shortest path algorithm proposed in [31]. The paths in $\mathcal{P}$ are sorted in a decreasing order of available bandwidth (denoted as B_p). The algorithm starts from the widest path and aggregates the available bandwidth from the paths in $\mathcal{P}$. The longest path in the selected paths is denoted as $\tilde{p}$ and used as a baseline of the path delay that contributes to the end-to-end delay. The current bandwidth allocated to R is denoted as $Resv$ and the current end-to-end delay is calculate as $C/Resv + pd_{\tilde{p}}$, which is compared with the transmission deadline D_R. The algorithm stops when $C/Resv + pd_{\tilde{p}} \leq D_R$. Note that the amount of bandwidth from p_i that contributes to $Resv$ is constrained by the parameter β, i.e., $B_{p_i} \cdot \beta$. The final value of $Resv$ is the actual bandwidth allocated to R.

Assuming the virtual topology has $|V|$ nodes and $|E|$ links, the worst case complexity for a loopless K-shortest path is calculated in at most $\mathcal{O}(K|V|(|E| + |V|log|V|))$ steps [31]. Alg. 2 considers all the K paths in path selection, therefore complexity of the proposed algorithm is $\mathcal{O}(K|V|(|E| + |V|log|V|) + K)$.

2.6.2 Algorithm for Real-time Streaming

Real-time streaming is commonly seen in cloud computing, telemedicine or remote visualization etc. To prevent video distortion,

Algorithm 2: Multipath routing algorithm for bulk data transfer

Input: Virtual topology $G(V, E, \beta)$; Connection request $R(s, d, C, D_R)$

Output: Multipath solution for R

Step 1: *Preprocessing*

Calculate K paths using K-shortest path algorithm [31] and sort the paths in a decreasing order of available bandwidth, $\mathcal{P} = \{p_i\}, i = 1, ..., K,\ B_{p_i} > B_{p_{i+1}}$;

Step 2: *Path selection and bandwidth allocation*

$Resv = 0$

for $i = 1, .., K,\ p_i \in \mathcal{P}$ **do**

  $Resv = Resv + B_{p_i} \cdot \beta$;

  $pd_{\tilde{p}} = max.\{pd_{p_1} ... pd_{p_i}\}$;

  if $(Resv \geq C/(D_R - pd_{\tilde{p}}))$ **then**

    break

  end

  Update bandwidth availability in $\mathcal{P}$

  Sort the $\mathcal{P}$ in a decreasing order of available bandwidth

end

Step3: Output the selected paths in the solution set $\mathcal{P}'$

data is extracted and decoded at the receiver within a deadline, which is also referred to as *jitter* constraint in video streaming [32]. A packet will only stay in the re- sequencing buffer for at most D_l (decoding deadline). For such applications, differential delay among multiple paths is critical.

A connection request for a real-time streaming application is denoted as $R(s, d, B_R, D_l)$, where (s, d) are source and destination, respectively. B_R is the required bandwidth. The decoding deadline D_l is used as the maximum differential delay constraint. The available

Algorithm 3: Multipath routing for real-time streaming

Input: Virtual topology $G(V, E, \beta)$; Connection request $R(s, d, B_R, D_l)$

Output: Multipath solution for R

```
1  //Serve the connection demand with single path routing first;
   otherwise use multipath routing.
2  Find the widest path p_0 between s and d;
3  if B_{p_0} · β ≥ B_R then
4  |  Output p_0 as the solution and break;
5  end
6  else
7  |  Calculate K paths by K-shortest path algorithm and sort in
   |  the decreasing order of the available bandwidth,
   |  P = {p_i}, i = 1, ..., K, B_{p_i} ≥ B_{p_{i+1}}; p̃ is the path with the
   |  largest delay in P.
8  end
9  //Path selection and bandwidth reservation.
10 for i = 1, .., K, p_i ∈ P do
11 |  P' = ∅; p̃ = p_i;
12 |  for p_j ∈ P do
13 |  |  if pd_{p_j} ≤ D_{p̃} then
14 |  |  |  p_j → P';
15 |  |  end
16 |  end
17 |  for p_k ∈ P' do
18 |  |  if pd_{p̃} − pd_{p_k} ≤ D_l then
19 |  |  |  Resv = Resv + B_{p_k} · β;
20 |  |  |  if Resv ≥ B_R then
21 |  |  |  |  break;
22 |  |  |  end
23 |  |  end
24 |  end
25 |  // Output the selected paths as the candidate solution.
26 |  for each path p_k in the selected paths do
27 |  |  // proportional bandwidth reservation on the selected
   |  |  paths;
28 |  |  t_{p_k} = B_R · B_{p_k} / Rev;
29 |  end
30 |  // compare the buffer size constraint;
31 |  if Σ_{p_k} t_{p_k} · (pd_{p̃} − pd_{p_k}) ≤ M_d then
32 |  |  Output solution and break;
33 |  end
34 end
```

buffer size in the final egress border node is used as a constraint in the path computation. The proposed algorithm for real-time streaming is shown in Alg. 3 [30].

The algorithm first tries to find a single path for request R by checking the widest path between s and d, while the parameter β is applied to constrain the maximum available bandwidth that can be allocated to the request. If the connection request can not be served with the widest path, the algorithm moves to find a multipath solution. K-shortest paths are computed [31] and ordered in $\mathcal{P}$ with a decreasing order of available bandwidth before further processing.

The path selection and traffic splitting strategy is developed around a guess of a path chosen from $\mathcal{P}$ as the longest path, denoted as $\tilde{p}$, and then composing a corresponding candidate path set, i.e., $\mathcal{P}'$, by choosing all the paths with a delay no larger than $\tilde{p}$. The mapping starts from choosing the widest path as the first guess and stops when the solution is found. For instance, assume path P_i is chosen to be $\tilde{p}$, the algorithm will compare the paths in $\mathcal{P}$ with p_i and put all the paths with delay that is not larger than pd_{p_i} in the candidate path set $\mathcal{P}'$. For instance, if four paths are calculated and ordered in $\mathcal{P} = \{p_1, p_2, p_3, p_4\}$, with the delay of each path as $pd_{p_1} = 4, pd_{p_2} = 2, pd_{p_3} = 3$, and $pd_{p_4} = 5$. When p_1 is chosen as the first guess to be $\tilde{p}$, the current candidate path set $\mathcal{P}'$ is composed as $\mathcal{P}' = \{p_1, p_3, p_2\}$. In each inner loop, the algorithm tries to find a solution from $\mathcal{P}'$. The differential delay constraint is first checked and is followed by the check of the bandwidth requirement. The current bandwidth allocated to R is $Resv$, which is constrained by β, i.e., $Resv = Resv + B_{p_k} \cdot \beta$, where p_k is the path that is being checked.

The traffic is split into the selected paths proportionally and the buffer size constraint is checked at the end. If all paths can be reserved simultaneously and the required buffer M_r is not larger than the available buffer M_d at the destination, the algorithm stops

and it returns the magnitude of the sub-flows. Otherwise, it goes to the outer loop and starts with another $\mathcal{P}'$. The algorithm stops only when a solution is found or all paths in $\mathcal{P}$ are exhausted.

Path selection and traffic splitting have time complexity of $\mathcal{O}(K^3)$. The complexity of finding the widest path in the first step is the same as the weighted Dijkstra shortest path, i.e., $\mathcal{O}(|E| + |V|log|V|)$; and the worst case time complexity of the loopless K-shortest path is $\mathcal{O}(K|V|(|E|+|V|log|V|))$ [31], where $|V|$ and $|E|$ represent the number of nodes and links in the virtual topology, respectively. Therefore, Alg. 3 has a complexity of $\mathcal{O}(K|V|(|E| + |V|log|V|) + K^3)$.

2.6.3 Evaluation

We evaluate Alg. 2 and Alg. 3 in a three-domain network, each with a NSFnet topology (as shown in Fig. 2.8). In each domain, the virtual topology is composed of the lightpaths computed in advance between the border nodes (marked with doted lines). The topology shown in Fig. 2.8 includes all critical elements of inter-domain service provisioning, i.e., source domain (Domain 1), destination domain (Domain 3) and transit domain (Domain 2). It is generic enough to evaluate the proposed algorithms. The proposed algorithms compute path(s) based on the constructed virtual topology.

Links connecting two domains, i.e., inter-domain links, are marked with bold lines and has a capacity of 40 Gbps. The capacity of the virtual links of Domain 1 and Domain 3 are assumed to be 10 Gbps. Given the fact that domain 2 is subject to higher load than the other domains, the capacity of the virtual links of Domain 2 is set to be 40 Gbps. For both bulk data transfer and real-time streaming applications, 10^5 connection requests arrive in a Poisson process and are uniformly distributed among all source and destination pairs.

In the results that follow, we first quantify the value of multipath routing in achieving a trade-off between scalability and resource efficiency by comparing it with single path routing. The performance

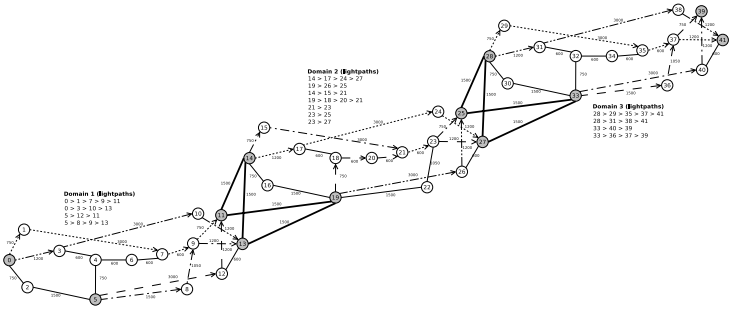


Figure 2.8: The three-domain topology used in performance evaluation

evaluation is focused on blocking probability and link utilization. We also analyze the average differential delay and the required buffer size of the proposed algorithms.

Bulk Data Transfer

Bulk data transfer is featured by the transitory connection time of each chunk, hence the proposed algorithm is evaluated against *connection arrival rate* which is defined as the average number of connection request arrivals per second. The blocking ratio is used as the performance index, which is defined as the percentage of rejected requests in all connection requests.

The size of the data chunk to be transferred on each connection is assumed to be 10 GB. A path can only be used by other connections after current transmission is finished. The maximum transmission duration is set to $8s$. With multipath routing, the data chunk can be transferred in a shorter duration, and network resource can be released earlier for other connection requests.

Reduction of blocking probability: As discussed earlier, using static virtual topology enables a scalable multi-domain service provisioning. However, it can lead to a high blocking probability when connec-

tion requests are arriving dynamically. It is especially phenomenal when only single path routing is allowed. Fig. 2.9 shows that, with single path routing, around 27% connection requests are blocked, even when less than one connection request arrives at the network per second. When the network receives two requests per second, more than 30% connections are blocked. With multipath routing, the blocking probability is significantly reduced. When less than one connection request arrives per second, the blocking probability is less than 10%. With a high arrival rate, e.g., two connection requests per second, 12% requests are blocked, which is much less comparing with single path routing. Lower blocking probability implies higher acceptance rate, leading to a increase of average link utilization. As shown in Fig. 2.10, the average link utilization is 6% higher with multipath routing when arrival rate is two requests per second, comparing with single path routing.

Load balancing: It is known that multipath routing can facilitate load balancing. However, due to the feature of bulk data transfer, i.e., very short duration, multipath routing does not necessarily lead to performance improvement in terms of load balance. As it is shown in Fig. 2.11, a small β can lead to a smaller blocking probability. However, it also increases blocking probability. Hence, the value of β needs to be carefully chosen. An example shown here is that there is a slight decrease of link utilization between $\beta = 0.9$ and $\beta = 0.8$ at $ArrivalRate = 2$ while block probability increases 6% as shown in Fig. 2.12. Finally, we studied the average number of paths that are used for bulk data transfer application. Fig. 2.13 shows that our algorithm takes two paths on average per connection regardless of the connection arrival rate.

Real-time Streaming

To evaluate the performance of multipath routing in case of real-time streaming applications, we define *network load* A (in Erlang) as $u*h$,

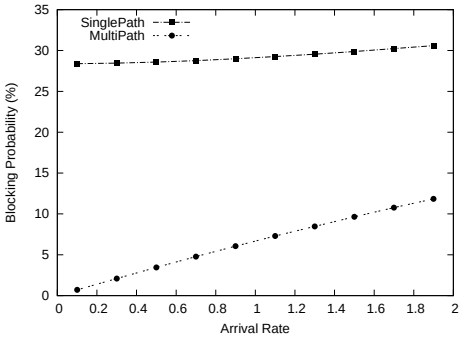


Figure 2.9: Blocking ratio of bulk data transfer; $\beta = 1$

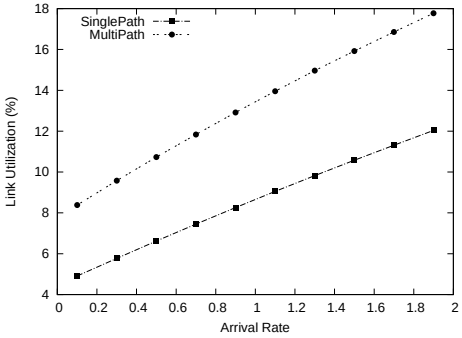


Figure 2.10: Link utilization in case of bulk data transfer; $\beta = 1$

where u is connection arrival rate and h is the mean connection holding time. The bandwidth required by the connection demands for the real-time streaming application varies from 250 Mbps to 8 Gbps and the number of connections is inversely proportional distribution of required bandwidth, i.e., the bandwidth requirement distribution of $250Mbps : 500Mbps : 1Gbps : 2Gbps : 4Gbps : 8Gbps$ leads to a proportion of the number of connection demands as $32 : 16 : 8 : 4 : 2 : 1$. *Bandwidth Blocking Ratio (BBR)* is used to evaluate the proposed

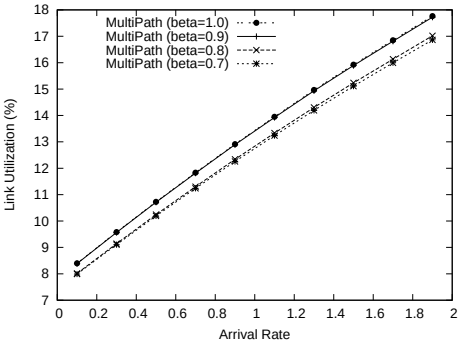


Figure 2.11: Link utilization in case of bulk data transfer vs. β

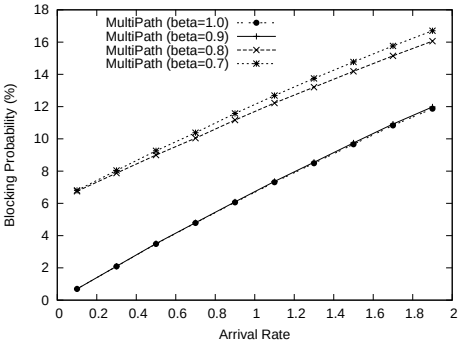


Figure 2.12: Blocking ratio of bulk data transfer applications vs. β

algorithm related to the real-time application, which is defined as the sum of bandwidth requested by the blocked connections divided by the total bandwidth required by all connections. The buffer size of all border nodes are assumed to be 10 MB.

Reduction of blocking probability: Fig. 2.14 shows that multipath routing can decrease the bandwidth blocking ratio. At first glance, these are intuitive results. However, a very slight increase in average link utilization can be observed in Fig. 2.15 (y axis in logscale),

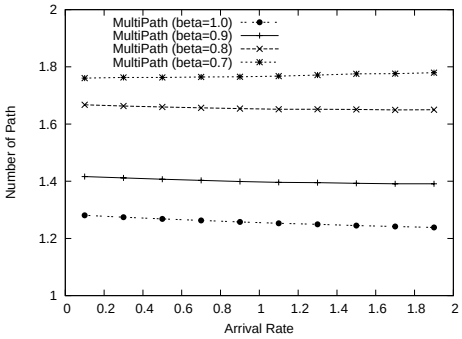


Figure 2.13: Average number of paths used in case of bulk data transfer vs. β

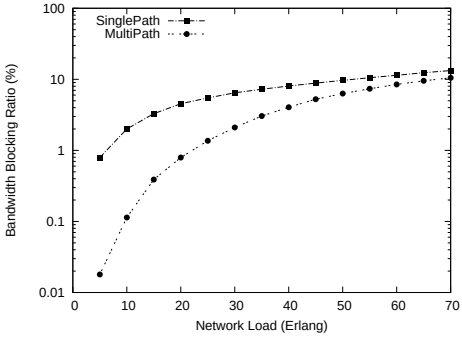


Figure 2.14: Bandwidth blocking ratio of real-time streaming; $\beta = 1$

especially when network load is high. For instance, average link utilization increases at most 2% at network load of 70 *Erlang*. It implies that the proposed algorithm can balance network load.

Load balancing: We further study the impact of the parameter β. As shown in Fig. 2.16, a strict constraint on the available bandwidth can lead to a significant increase in the bandwidth blocking ratio. 10% bandwidth blocking ratio is observed for $\beta = 0.7$ even when

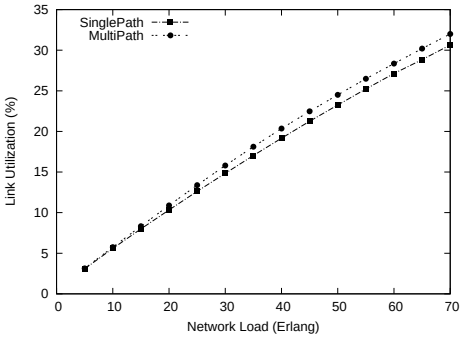


Figure 2.15: Link utilization in case of real-time streaming; $\beta = 1$

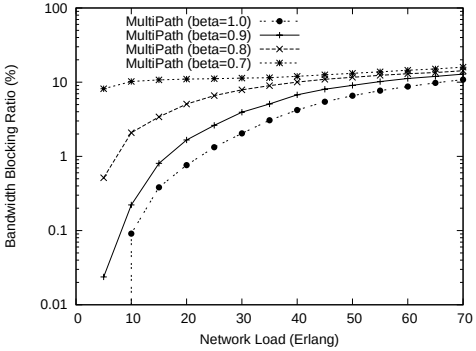


Figure 2.16: Bandwidth blocking ratio of real-time streaming vs. β

the network load is lower than 20 *Erlang*. Inappropriate values of β can limit the benefits of using multipath routing. However, a carefully chosen β can achieve comparable bandwidth blocking ratio values while balancing the traffic. For instance, a trade-off solution is obtained with $\beta = 0.9$ in the case study shown in Fig. 2.17.

Cost of using multipath routing: The cost of multipath routing is the buffer required for compensating differential delay. Please note that all the solutions found by our algorithm respect the available

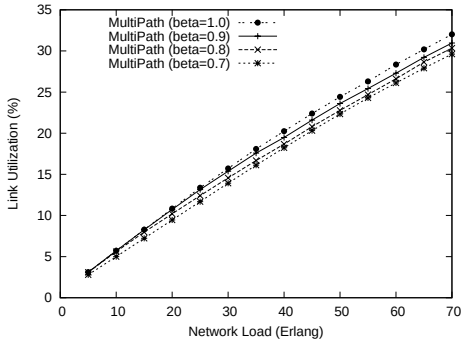


Figure 2.17: Link utilization in case of real-time streaming vs. β

buffer size constraint. We aim to quantitatively illustrate the buffer size required in the multipath solutions in a multi-domain network. Fig. 2.18 shows the average differential delay of multipath routing with different values of β. It can be seen that small values of β lead to more frequent traffic splitting, which results in larger average differential delays. As a result, the buffer size requirements also increase, as shown in Fig. 2.19. However, the average differential delay of multipath solutions are still small, generally less than $0.008ms$ when network load is below 70 $Erlang$. The average buffer size required by the proposed solutions are generally less than 1.5KB, which is also comparably low.

2.7 Protocol Extensions and Implementation

The PCE and relevant protocols are designed to support single path routing. Extending PCE with multipath computation capability is particularly interesting for high-speed Ethernet transmission where parallelization is a trend [2]. While 100GE serial interfaces are still in lab-trial phase, parallel transmission over multiple interfaces over low speed channels is a valid solution [33]. It has been proposed

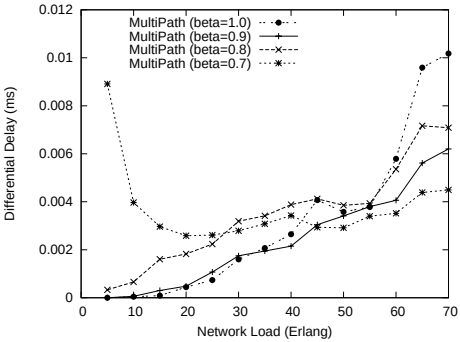


Figure 2.18: Average differential delay of real-time streaming connections vs. β

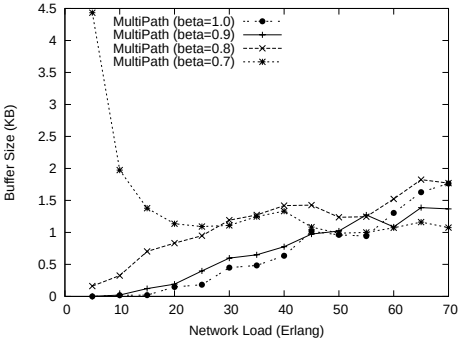


Figure 2.19: Average buffer size used by real-time streaming connections vs. β

to use multiple MAC layouts ranging from 27.5x4Gbps, 10x10Gbps, 3x40Gbps and so on, utilizing low speed optical channels [2]. This section presents protocol extensions to enable multipath computation with the PCE. The proposed extensions are implemented using an open-source PCE emulator [15]. The implementation details can be found in [16].

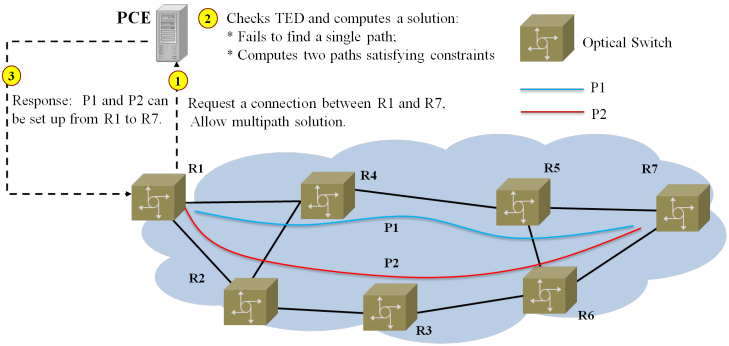


Figure 2.20: Extend PCE to Support multipath routing

To better understand the scenario of using PCE for multipath computation, we show an example in Fig. 2.20. The Path Computation Client (PCC), i.e., R1 here, sends a path computation request to the PCE with QoS constraints (Step1). In this example, the maximal acceptable differential delay (DD) is included as a specific constraint of multipath routing. Upon receiving the request, the PCE checks the available resources in its TED and computes a single path solution or multipath solution for the connection request, depending on the resource availability in the network as shown in Step 2. In this example, the PCE fails to find a single path between R1 and R7. When multipath routing is enabled, the PCE computes two paths, namely P1 and P2 and sends Explicit Route Object (ERO) of the computed paths to R1 (Step3).

2.7.1 Multipath Extensions in PCEP Protocol

As standardized in [12], the PCEP protocol has defined seven messages. To enable multipath routing in PCE, two messages need to be modified, namely, Path Computation Request message (PCReq) and Path Computation Response message (PCRep) as shown in Fig. 2.21

and Fig. 2.22, respectively.

<PCReq Message> :: = <Common Header>
< Multipath>
<End-points>
<Metric-list>
Where <Metric-list> ::= <Maximal acceptable differential delay>
<End-to-end delay>
<Bandwidth>

Figure 2.21: PCEP request message with multipath extension

<PCRep Message> :: = <Common Header>
<End-points>
<Path-list>
<Metric-list>
Where <Path-list> :: = [<ERO><Bandwidth>]
Where <Metric-list> ::= <Maximal acceptable differential delay>
<End-to-end delay>

Figure 2.22: PCEP reply message with multipath extension

The multipath extension in the PCReq message utilize the field *Metric-list* to specify the multipath related constraint, i.e., *Maximal acceptable differential delay*. The extensions in the PCRep includes *Path-list* and *Metric-list*. The *Path-list* contains the information of ERO and allocated bandwidth of computed paths. Upon receiving the multipath computation request, PCE initiates path computation based on the resource information in its TED. In case the PCE can compute a single path for the connection request, the *Path-list* only contains one path. Otherwise, the PCE computes a multipath solution. If a set of paths can be found that fulfill the QoS requirements, including bandwidth, end-to-end delay and maximal differential delay, PCE returns a PCRep message with *Path-list*, specifying a set of EROs and bandwidth assigned on each path.

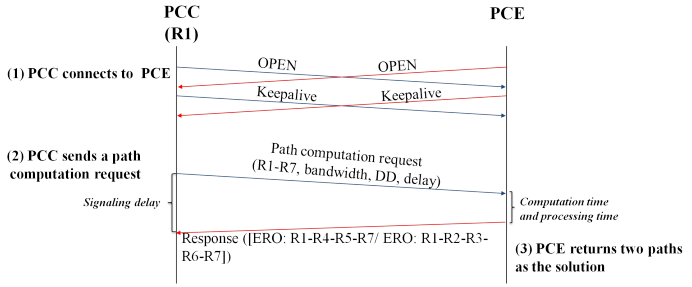


Figure 2.23: PCE signaling for multipath routing; Note that messages such as *Close* and *Error* are not shown in this signaling flow

2.7.2 PCE Signaling for Multipath Computation

The signaling flow of the PCE with multipath extensions is shown in Fig. 2.23. It is in line with the basic signaling rules specified in RFC5440 [12]. The signaling flow is described as follows:

1. PCC and PCE exchange the *Open* and *Keepalive* messages to open the session. If the *Open* and *Keepalive* messages are successfully exchanged between PCC and PCE, the PCC is connected and ready to send path computation requests to the PCE.

2. In the path computation request, i.e., PCReq, PCC specifies the source and destination of the connection and QoS constraints, such as bandwidth and delay as well as the maximal acceptable differential delay (DD).

3. If the PCE succeeds to compute a set of paths based on the embedded multipath routing algorithm, satisfying all the given QoS constraints, it returns a PCRep message that contains a set of EROs and allocated bandwidth on each path.

In the example shown in Fig. 2.20, the Path-list contains two paths, i.e., P1 and P2. Therefore, the EROs included in the PCRep message are R1-R4-R5-R7 and R1-R2-R3-R6-R7 with bandwidth allocated on each path is also specified.

2.8 Summary

This chapter outlined the the challenges and issues of multipath routing and presented novel solutions from all perspectives, including novel algorithm design, protocol extension and implementation and proposals of novel service provisioning scheme for multi-domain scenarios.

In contrast to the conventional multipath routing where flow-base traffic distribution was applied, we focused on data-intensive applications and formulated a new multipath routing problem in optical networks for large unicast flows. We proposed an optimization model for optimal path selection and traffic splitting such that a trade-off between the bandwidth and memory cost can be achieved. We showed that multipath routing can be used to achieve a scalable inter-domain service provisioning and presented two novel multi-domain multipath computation schemes, namely, *segmental multipath computation* and *end-to-end multipath computation*. We proposed heuristic algorithms tailored for two representative data-intensive applications: bulk data transfer and real-time streaming. Finally, we completed the study with protocol extensions and implementation of multipath routing using an open-source PCE emulator.

3
From Multipath Routing to Parallel Transmission

3.1 Introduction

While network service provisioning paradigm is shifting from single path routing to multipath routing driven by data-intensive applications, end-systems have also gone through the transformation from serial to parallel. A prominent example is the emerging high-speed Ethernet. Instead of using high-speed serial interfaces, 40 Gbps Ethernet (40GE) and 100 Gbps Ethernet (100GE) utilize parallel optics to split traffic across multiple lanes with lower rates, which is referred to as *Multi-lane Distribution* (MLD) [2]. To this end, it has been specified that 40 GE and 100 GE can utilize 4 and 10 parallel lanes, respectively, with each lane running at 10.3125 Gbps [2].

The parallelism in the end-systems of high-speed Ethernet is driving the need for *parallel transmission* in optical networks. Most present optical networks are designed to support transmission rates of 10Gbps, or 40Gbps in some recently deployed networks. While the high-speed Ethernet may scale up to 400 Gbps or even beyond, parallel transmission over multiple wavelengths/fibers is more economical and backward compatible than its high-speed serial counterpart. Moreover, optical reach is known to decrease with the increase of serial transmission bit rate, due to transmission non-linear effects, group velocity dispersion and polarization mode dispersion. Parallel

transmission can benefit from using low rate channels to achieve a long optical reach.

This chapter presents the first study on feasibility of parallel transmission in OTN/WDM networks to support high-speed Ethernet [3]. We propose a novel architecture which integrates the parallelism in Ethernet layer and parallel transmission in OTN/WDM networks. The proposed architecture is in compliance with standards of both Ethernet and optical layers, i.e., IEEE 802.3ba and ITU-T G.709. After designing the system, we combine the parallelization in the end-system with multipath routing in the optical networks and model a Multipath Routing and Wavelength Assignment (MRWA) problem tailored for high-speed Ethernet parallel transmission in OTN/WDM networks, which has not been studied to this end. We propose an ILP based optimization model with consideration of all possible buffering availability, including electronic buffer and Fiber Delay Line (FDL) based optical buffer. To counter the complexity issue of ILP, we utilize an Evolutionary Algorithm (EA) based optimization approach to find optimal or near-optimal solutions in a reasonable time. Finally, we also propose a mechanism for parallel transmission without requiring buffers, referred to as *bufferless parallel transmission*.

3.2 Supporting Publications

1. X. Chen, A. Jukan, "Optimized Parallel Transmission in OTN/WDM Networks to Support High-Speed Ethernet with Multiple Lane Distribution (MLD)", IEEE/OSA Journal of Optical Communications and Networking (JOCN), Volume 4, Issue 3, March 2012, PP. 248-258.

2. X. Chen, A. Jukan, A. Gumaste, "On the Usage of FDLs in Optical Parallel Transmission to Support High Speed Ethernet," in International Conference on Optical Network Design and Modeling (ONDM), April 2012.

3.3 High-speed Ethernet Parallel Transmission

3.3.1 Data Framing and Parallelization

To better understand the parallelism in the end-system of Ethernet layer, we show the serial to parallel conversion of Ethernet traffic in Fig. 3.1. The aggregated native Ethernet frames at a high-speed, e.g., 40 Gbps or 100 Gbps, are scrambled and regrouped into data blocks of same size, i.e., 64b. Each block is inserted with two bits as overhead to indicate if the block carries pure payload or control information. The data blocks are distributed to parallel virtual Ethernet lanes in Multiple Lane Distribution (MLD) layer, in a round robin fashion [2]. Traffic from each lane is routed on a path in the optical network.

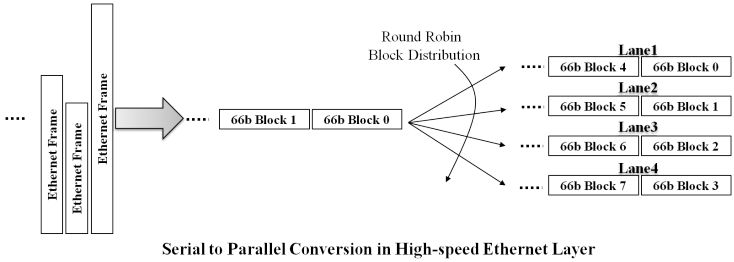


Figure 3.1: Serial to parallel conversion of high-speed Ethernet traffic according to IEEE 802.3ba [2]

3.3.2 Reference Architecture

Fig. 3.2 shows the reference architecture of high-speed Ethernet specified in IEEE 802.3ba. The electronic module includes a packet interface where the aggregated 40G/100G IP traffic is injected. The Ethernet frames encapsulated in the MAC layer will be scrambled

and formatted into data blocks of same size using 64b/66b line coding in Physical Coding Sublayer (PCS). Data stripping and parallelization are also implemented in the PCS module. A lane marker, which is a special 66b block, is inserted into each lane every 16,382 blocks as the lane marker for synchronization in the PCS module of the receiver [2].

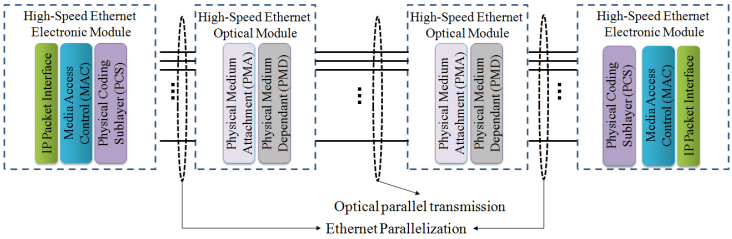


Figure 3.2: High-speed Ethernet architecture [2]

The Ethernet lanes are mapped onto optical channels which can be multiple wavelengths in a single fiber or diverse fibers. It is not necessary that the number of Ethernet lanes is the same as the number of optical channels. For instance, 10 Ethernet lanes from 100GE may be multiplexed into four optical optical channels with each channel of 25 Gbps [1]. As a result, the parallel optics systems are used in high-speed Ethernet cabling solution. For instance, 40GE uses a solution of 12-fiber cabling, with four dedicated fibers for transmitting and receiving, respectively. The remaining are backup dark fibers [2].

To address the differential delay issue (which is referred to as *skew* in IEEE 802.3ba), it is clear that an external memory is required. However, it is challenging to meet the speed and throughput requirements for high-speed Ethernet transmission. This processing challenges are compounded by an ever-increasing speed and the need for lowering cost.

[1] For the detailed mapping schemes, please refer to IEEE802.3ba [2].

3.3.3 Parallel Transmission in Optical Networks

As shown in Fig. 3.2, optical parallel transmission is applied between optical modules at source and destination, respectively. The traffic from each virtual Ethernet lane can be routed independently in optical networks.

Fig. 3.3 shows an example of 40GE parallel transmission over a WDM network. Assuming that capacity per wavelength is 10 Gbps, it requires four wavelengths between two high-speed Ethernet switches. However, existing Routing and Wavelength Assignment (RWA) algorithms are designed to find and assign a single wavelength. New algorithms are required for WDM networks, which can find multiple paths and assign wavelengths accordingly. We referrer it to as *Multipath Routing and Wavelength Assignment (MRWA)* problem and present solutions in the following section.

3.4 Design and Modeling of Parallel Transmission in OTN/WDM Networks

3.4.1 Reference Architecture

Fig. 3.4 shows the proposed novel architecture for high-speed Ethernet parallel transmission in OTN/WDM networks. The OTN layer, as specified in ITU-T G.709 [17], is an electronic layer which distributes high-speed data into multiple OTU-k containers. The OTN information hierarchy enables the mapping between Ethernet lanes and wavelengths, and Optical Virtual Concatenation (OVC) protocol in OTN layer facilitates parallel transmission in WDM layer.

The case shown in Fig. 3.4 is one-to-one mapping, where four wavelengths are used for four OTU-k channels. We assume all-optical transmission; thus each path uses the same wavelength from source to destination. Here, the capacity per wavelength is 10 Gbps and four wavelengths are used to support 40 Gbps Ethernet trans-

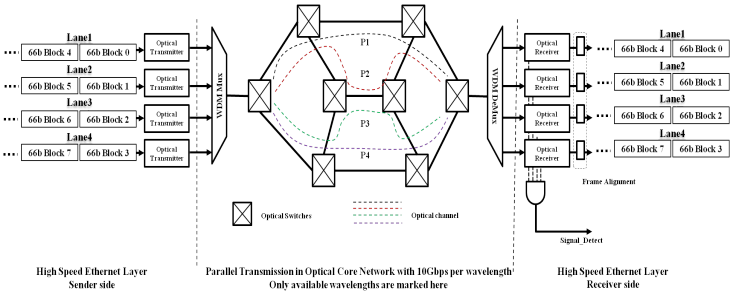


Figure 3.3: High-speed Ethernet parallel transmission over WDM

mission. However, the same architecture can also support high-speed Ethernet with m Ethernet lanes using n wavelengths, where $n \neq m$. Assume HSE-1 and HSE-2 are 100 Gbps Ethernet networks and the capacity of a wavelength is 40 Gbps, it requires three wavelengths to support the data transmission between HSE-1 and HSE-2. Both cases, i.e., $n = m$ and $n \neq m$, results in a bandwidth requirement in form of the number of wavelengths.

Given the assumption of all-optical transmission, electronic processing only happens at source (s) and destination (d) in upper layers (Ethernet and OTN). In other words, there is no wavelength conversion along the path. The same wavelength is assigned on all fiber links along the path. Each Ethernet transmitter can use at least one wavelength. The number of wavelengths required for parallel transmission of a single connection depends on the capacity of logical containers (ODU-k) and and capacity per wavelength.

3.4.2 Challenges and Issues

As mentioned earlier, parallel transmission in OTN/WDM networks is challenged by the so-called *differential delay* issue (skew) caused by using multiple wavelengths. Refer to the reference architecture shown in Fig. 3.4, and let us assume that four wavelengths are al-

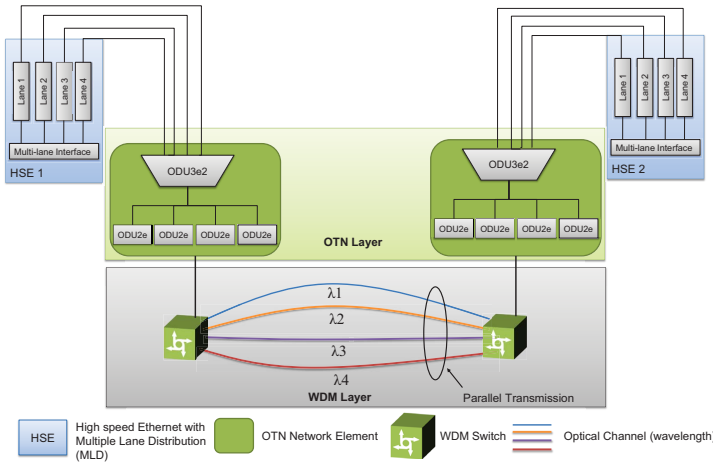


Figure 3.4: Parallel transmission in OTN/WDM networks to support high-speed Ethernet: Reference architecture [3]

located in different fibers. Each wavelength experiences different end-to-end delay, resulting in the different arrival of traffic at the destination node. However, due to the differential delay, the order of data blocks turns to be $1, 3, 2, 4...$ (marked in red), after transmission over $\lambda_1, \lambda_2, \lambda_3$ and λ_4, as shown in Fig. 3.5.

Electronic buffering is an effective mechanism for skew compensation in parallel transmission. The buffer size required to align traffic from diverse paths is closely related to the cumulative differential delay in paths used for the connection. Assume a set of paths, $\mathcal{L} = \{p_i, i = 1, 2...k\}$, are computed for a connection request R and $\tilde{p}$ is the one with the highest delay in $\mathcal{L}$, buffer required to ensure

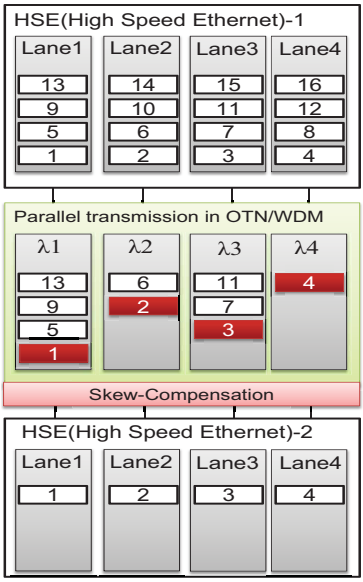


Figure 3.5: Differential delay issues of parallel transmission in OTN/WDM networks [3]

in-order delievery is calculated as in Eq. (3.1) [34]:

$$M_R = \sum_{p_i, i=1,2\ldots k} C_i \cdot (d_{\tilde{p}} - d_{p_i}) \tag{3.1}$$

where C_i is the capacity of path p_i, i.e., the capacity per wavelength in WDM layer and d_p is the delay of path p. For example, in Fig. 3.5, $k = 4$ and $\tilde{p} = \lambda_4$.

The differential delay in the parallel transmission should not exceed the compensation capability of the system, i.e., the electronic buffer equipped in the OTN layer. It is currently open to research whether the parallel transmission in OTN/WDM networks can be implemented without large electronic buffers. Moreover, optimal so-

lutions to parallel transmission are hard to obtain in real- time due to the complexity of the problem.

3.4.3 ILP Optimization Model

In this section, an optimization model based on ILP is presented for MRWA problem [3]. A WDM network is represented as $G(V, E)$, where V is the set of nodes and E is the set of links. We assume all links in the network have the same number of wavelengths, denoted as $\mathcal{W}$. The set of available wavelengths on link e is denoted as W_e with $W_e \subseteq \mathcal{W}$. A connection request is represented as $R(s, d, r)$, where s and d are source and destination, respectively. r is the bandwidth requirement in the number of wavelengths. The link delay of e is denoted by LD_e and $LD_e \in \mathbb{N}$. Path delay is denoted as pd_p and $pd_p = \sum_{e \in p} LD_e$. The computed lightpaths are placed in a path set, denoted as $\mathcal{P}$. The variables of the ILP model are as follows:

- x_p : Binary variable denotes if p is found and placed in $\mathcal{P}$.
- $x_{p,w}$: Binary variable denotes if a wavelength w on path $p \in \mathcal{P}$ is used.
- $x_{p,e}$: Binary variable denotes if the edge $e \in E$ is on the path $p \in \mathcal{P}$.
- $o_{p,p'}$: Binary variable denotes if $p \in \mathcal{P}$ and $p' \in \mathcal{P}$ share at least a link.
- $pd_{p,w}$: Integer variable denotes the delay of the path $p \in \mathcal{P}$ using wavelength $w \in \mathcal{W}$. It is 0 if the wavelength w is not used on path p.
- md : Integer variable which is the maximal delay in current solution.

- $x_{p,e,w}$: Binary variable denotes if p uses wavelength w on e.

The number of used wavelengths on a link e is calculated as $\sum_{p\in\mathcal{P},w\in\mathcal{W}} x_{p,e} \cdot x_{p,w}$, which is non-linear and can not be used directly in the ILP optimization. We define a new variable as $x_{p,e,w}$ to denote that the wavelength w on link e is used by the computed path p. Hence, the total number of used wavelengths on link e can be calculated as $\sum_{p\in P,w\in W} x_{p,e,w}$, with $x_{p,e,w} = x_{p,e} \cdot x_{p,w}$. The value of $x_{p,e,w}$ is determined by a non-linear function, which can be linearized as follows:

$$x_{p,e} + x_{p,w} - x_{p,e,w} \leq 1 \tag{3.2}$$

$$x_{p,e} - x_{p,e,w} \geq 0 \tag{3.3}$$

$$x_{p,w} - x_{p,e,w} \geq 0 \tag{3.4}$$

The objectives considered in the optimization model are:
1) **Resource Consumption Minimization**, as defined in Eq. (3.5), is to minimize the overall optical resources allocated to the connection request R. As a result, the number of connections that an OTN/WDM network can accommodate is maximized. In Eq. (3.5), Link delay (LD_e) is used as a weight in the objective function such that the shortest paths are preferred in the optimization.

$$\text{Minimize} \sum_{p\in P,e\in E,w\in W} LD_e \cdot x_{p,e,w} \tag{3.5}$$

2) **Max. Differential Delay Minimization**, as defined in Eq. (3.6), is to find a path set which has minimal differential delay between the longest and shortest path in the path set. As a result, the required electronic buffer at the destination node can be minimized.

$$\text{Minimize} \quad \text{Max}_{p\in P,w\in W}\{md - pd_{p,w}\} \tag{3.6}$$

The optimization model subjects to the following constraints:
***Routing constraints*:** Eq. (3.7) is defined to ensure that incoming

traffic is equal to outgoing traffic at every intermediate node, except for source and destination nodes. Eq. (3.8) and Eq. (3.9) constrain that traffic enters the network at the source node and leaves at the destination node. Eq. (3.10) and Eq. (3.11) are defined to eliminate the possible loops on a path by restricting that a node along a path can only have at most one predecessor and one successor.

$$\forall p \in P, \tilde{v}, v \in V, v \neq s, d : \sum_{e=(\tilde{v},v)\in E} x_{p,e} = \sum_{e=(v,\tilde{v})\in E} x_{p,e} \tag{3.7}$$

$$\forall p \in P, \tilde{v} \in V : \sum_{e=(\tilde{v},d)\in E} x_{p,e} = x_p \tag{3.8}$$

$$\forall p \in P, \tilde{v} \in V : \sum_{e=(s,\tilde{v})\in E} x_{p,e} = x_p \tag{3.9}$$

$$\forall p \in P, \tilde{v}, v \in V : \sum_{e=(v,\tilde{v})\in E} x_{p,e} \leq 1 \tag{3.10}$$

$$\forall p \in P, \tilde{v}, v \in V : \sum_{e=(\tilde{v},v)\in E} x_{p,e} \leq 1 \tag{3.11}$$

***Wavelength assignment constraints*:** Eq. (3.12) determines the value of $x_{p,w}$. It sets $x_{p,w}$ to zero when wavelength w in link e is not available for p, i.e., $w \setminus W_e$. The binary variable $o_{p,p'}$ is defined to denote if two paths p and p' have at least one common link. The value of $o_{p,p'}$ is determined by Eq. (3.13) and Eq. (3.14). When path p and p' share at least one fiber link, $o_{p,p'}$ equals to 1. When $x_{p,e}$ and $x_{p',e}$ both take a value of 1, the link e is identified as a shared link by p and p'. On the shared link e, a wavelength can only be assigned to one path for a connection R. With wavelength continuity constraint in the WDM layer, $x_{p,w}$ and $x_{p',w}$ can not take a value of one at the same time when $o_{p,p'} = 1$. Eq. (3.15) is used to avoid the conflict in wavelength assignment, and Eq. (3.16) ensures that wavelength assignment is only implemented when the path p is

selected for the connection demand.

$$\forall p \in P, e \in E, w \in W \setminus W_e : x_{p,e} + x_{p,w} \leq 1 \tag{3.12}$$

$$\forall p, p' \in P, p \neq p', e \in E : x_{p,e} + x_{p',e} - o_{p,p'} \leq 1 \tag{3.13}$$

$$\forall p, p' \in \mathcal{P}, e \in E : o_{p,p'} \leq \sum_{e} x_{p,e} \cdot x_{p',e} \tag{3.14}$$

$$\forall p, p' \in P, p \neq p', w \in W : x_{p,w} + x_{p',w} + o_{p,p'} \leq 2 \tag{3.15}$$

$$\forall p \in P, w \in W : x_p - x_{p,w} \geq 0 \tag{3.16}$$

Bandwidth requirement constraint: Eq. (3.17) ensures that the number of wavelengths assigned to all the paths for connection demand R should be equal to the bandwidth requirement r, i.e.,

$$\sum_{p \in P, w \in W} x_{p,w} = r \tag{3.17}$$

3.4.4 Skew Compensation with Electronic Buffer

Considering the worst case scenario in parallel transmission, i.e., traffic from all paths except for the one with largest delay needs to be buffered, the electronic buffer required by the parallel transmission is calculated according to Eq. (3.1), which depends on the largest delay in the candidate path set.

Buffer constraints: The value of md needs to be calculated in order to find the best set of paths that can yield a minimum buffer requirement. The delay of a path is defined in Eq. (3.18). The value of md in the current path set is defined in Eq. (3.19). Eq. (3.20) defines that the required buffer, denoted as M_r should not exceed the available buffer M_D at destination node.

$$\forall p \in P, w \in W : pd_{p,w} = \sum_{e} LD_e \cdot x_{p,e,w} \tag{3.18}$$

$$\forall p \in P, w \in W : md \geq pd_{p,w} \tag{3.19}$$

$$\forall p \in P, w \in W : M_D \geq M_r, M_r = \sum_{w} \sum_{p} C \cdot (md - pd_{p,w}) \tag{3.20}$$

3.4.5 Skew Compensation with Optical Buffers

Another method to compensate differential delay in WDM networks is to utilize the Fiber Delay Lines (FDLs) to prolong the shorter paths [5]. Figure 3.6 shows an example of FDL buffer which was proposed in [4].

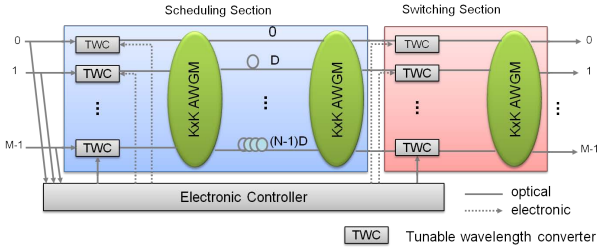
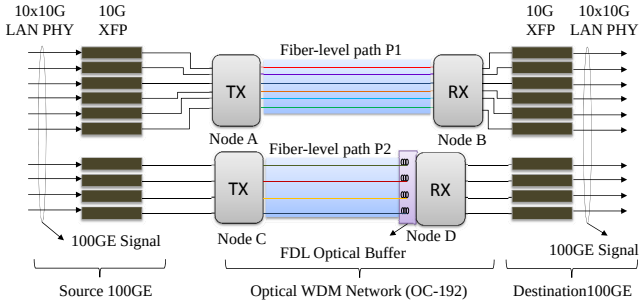


Figure 3.6: An illustrative FDL buffer architecture [4]

The granularity of each delay unit is denoted as D, and each fiber delay line can delay $n \cdot D$ units, $n = 0, 1, 2...N-1$. Traffic can not be circulated in the FDL buffer once it exits the line. The $K \times K$ Arrayed-Waveguide Grating (AWG) multiplexers enable the traffic from any input to be routed to any delay line with required delay units. Each fiber delay line can only be used by a wavelength at one time. The architecture shown in Fig. 3.6 covers all principal features of a FDL buffer. We therefore use it as a reference to formulate FDL constraints.

Considering the high-speed Ethernet as the representative application, Fig. 3.7 illustrates how FDLs can be used as optical buffers. Here, 10×10GE signals are distributed to two fibers, each has multiple wavelengths. Denote the two fiber-level paths as fp_1 and fp_2 between node pair C and D is shorter than fp_1 between node A and B. Traffic routed on all wavelengths in fp_2 needs to be buffered at the receiver side in order to re-sequence the frames/packets to be the right order. FDLs "buffer" optical packets by adding extra de-



**Only available wavelengths on fiber links are shown, and capacity per wavelength is 10Gbps; end-to-end delay between node A and B is larger than the delay between C and D.*

Figure 3.7: Use FDLs in optical parallel transmission [5]

lay to the shorter path with a selected fiber delay line [4]. In this example, FDLs at node D are utilized to prolong the fp_2. However, a FDL buffer can only provide discrete delay (buffer) units, which is decided by the length of fiber per delay unit and presents a considerable practical limitation in its deployment. The new variable defined for using FDLs in optical parallel transmission is defined as follows:

- n_{p,w,v,d_i} : Binary variable that denotes if the line d_i (the delay of line i is $i \cdot D$) in the FDL buffer in node v is used by $p \in P$ assigned with wavelength $w \in W$.

Optical buffer (FDLs) constraints: Integration of FDLs changes the end-to-end delay of the path. Assume each FDL buffer has N fiber delay lines, the delay of a path with consideration of FDLs is defined in Eq. (3.21). The constraints for optimally utilizing available fiber delay lines are defined as follows. Eq. (3.22) ensures that a fiber delay line can only be assigned to one wavelength at a time. Eq. (3.23) ensures that one wavelength can use the FDL buffer at

most once in one node, which is a restriction imposed by the FDL buffer architecture.
$\forall p \in P, w \in W, e \in E$:

$$pd_{p,w} = \sum_{e \in E} \{LD_e \cdot x_{p,e,w} + \sum_{i=\{1,..,N\}} (i \cdot D) \cdot n_{p,w,v,d_i} \cdot x_{p,e,w}\} \quad (3.21)$$

$$\forall v \in V p \in P, w \in W : \sum_{p \in P} \sum_{w \in W} n_{p,w,v,d_i} \leq 1 \quad (3.22)$$

$$\forall p \in P, w \in W, v \in V : \sum_{i=\{1,..,N\}} n_{p,w,v,d_i} \leq 1, i = \{1,..,N\} \quad (3.23)$$

3.4.6 Bufferless Parallel Transmission

Buffer requirements for skew compensation might hinder practical deployment of parallel transmission in WDM networks. This section presents a solution for parallel transmission without requiring buffer, referred to as *bufferless parallel transmission* [3]. An example is shown in Fig. 3.8 based on the reference architecture shown in Fig. 3.4. The original data stream is denoted as a sequence of frames, i.e., $F_1, F_2, ..F_n$. Traffic from HSE-1 is distributed over four Ethernet lanes and an Ethernet lane is mapped into an optical channel. Assume the data rate of one wavelength in the OTN/WDM network is C, the time for sending a Frame into an optical path is F/C. If delay of path 2 (λ_2) plus the time for sending a frame into path 2 is no less than the delay of path 1 (λ_1), it does not need buffering for traffic on path 1, i.e., $pd_2 + F/C \geq pd_1$. If the differential delay between pd_3 and pd_2 and between pd_4 and pd_3 satisfies the same constraint, all frames can be received in a correct order without buffering.

Bufferless parallel transmission constraint: Assume paths computed for parallel transmission are placed in a $\mathcal{P}$ and $\mathcal{P} = \{p_1, p_2, ..., p_i, ...\}$, where i is the order of round-robin distribution, i.e., if the first frame is distributed into p_i, the second frame will be distributed into p_{i+1},

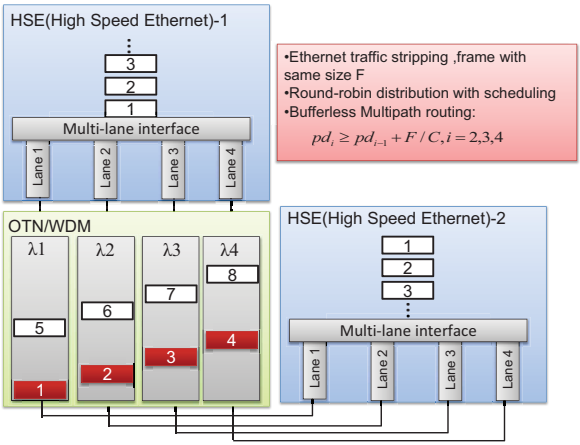


Figure 3.8: Bufferless parallel transmission with round-robin frame distribution

the bufferless parallel transmission needs to fulfill the constraint defined in Eq. (3.24).

$$\forall p_i \in P, w \in W, v \in V : pd_{p_{i+1},w} \geq pd_{p_i,w} - F/C \tag{3.24}$$

Proof: Refer to the example shown in Fig. 3.5, assume the first frame F_1 is sent out to p_1 at the time $t = 0$, it arrives at node D at t_1, which is:

$$t_1 = F/C_1 + pd_{p_1} \tag{3.25}$$

where F/C_1 is the time for sending the entire frame into the path p_1. The second frame F_2 is sent out after the F_1 is processed. Therefore, we can derive the time that F_2 arrives at the destination D as

$$t_2 = F/C_1 + F/C_2 + pd_{p_2} \tag{3.26}$$

When $t_2 \geq t_1$, the traffic arrives in order at the destination, which

does not require any buffer. Hence, we have:

$$F/C_2 + pd_{p_2} \geq pd_{p_1} \tag{3.27}$$

After K frames have been sent int K paths, the frame F_{K+1} will be sent to the path p_1 according the round robin frame distribution model. The bufferless property holds if and only if $t_{K+1} \geq t_K$. Therefore, we reach the same rule as shown in Eq. (3.29), i.e.,

$$\sum_{i=1,2,..,K+1} F/C_i + pd_{p_1} \geq \sum_{i=1,2,..,K} F/C_i + pd_{p_K} \tag{3.28}$$

We can derive the constraint of mulpath bufferless parallel transmission, i.e.,

$$pd_{p_{i+1}} \geq pd_{p_i} - F/C_{i+1} \tag{3.29}$$

Given the same data rates of wavelengths in a WDM network and the same frame size in the Ethernet layer, Eq.(3.24) can be derived.

3.4.7 Problem Size of the Optimization Model

The worst case complexity of an ILP formulation is known to be exponential in $O(2^n)$, where n is the number of boolean variables. The presented ILP based optimization for parallel transmission has an exponential complexity with n in $O(|P| \cdot (|P| + |E| \cdot |W|))$. When the FDL constraints are considered, the complexity of the ILP optimization problem increases, which is $O(2^n)$ where n is in $O(|P| \cdot (|P| + |E| \cdot |W| + |W| \cdot |V| \cdot |N|))$. This complexity of the ILP prohibits the applicability of the optimization model in practice.

For instance, the number of variables $x_{p,e}$ and $x_{p,e,w}$ grow rapidly with increasing network size. When FDLs are considered, the number of variable n_{p,w,v,d_i} that grows with increasing network size further enlarges problem size. This effect becomes more pronounced with an increase of the number of wavelengths per fiber. Take a simple example, for a network of size $|V| = 15$, $|W| = 16$, there are $|V| \cdot |V - 1| = 210$ node pairs. For each node pair, there are $|P| \cdot |W|$

instances of $x_{p,w}$. Assume four paths are used by the connection, i.e., $|P| = 4$, we have 13440 $x_{p,w}$ variables. Other variables can be calculated in a similar way. Thus, the total number of variables is very high even in small networks.

The problem size can be reduced by pruning the variables. A common method in the literature is to compute a set of paths in advance which are used as an input to the ILP optimization. For instance, if 10 paths are computed in advance and 4 paths are selected as a solution from the optimization, therefore, the number of $x_{p,w}$ variables is reduced to be $C_{10}^{4} \cdot |W| = 3360$. The same reduction can be obtained on the number of other variables. However, the solutions are limited in the pre-computed path set and the complexity of the path computation should also be considered. Moreover, the optimization model may fail to yield a solution from the paths computed in advance.

Another approach to address the complexity issue of the ILP optimization is Evolutionary Algorithm (EA) which can obtain a near-optimal or best solution in a reasonable time. In the next section, a genetic algorithm is introduced which utilizes an evolutionary optimization algorithm to solve the presented optimization model for parallel transmission.

3.4.8 Evolutionary Optimization

The Evolutionary Algorithm (EA)-based optimization (Alg. 4) starts with encoding the binary variables from the ILP optimization formulation. We use the encoding technique proposed in [35] to encode the binary variables into chromosome space and search feasible solutions based on backtracking method. Multi-objective optimization is applied in the evolutionary approach. Therefore, two objective functions defined in Eq. (3.5) and Eq. (3.6) are optimized simultaneously till the point where any further optimization can lead to a degraded optimality of another objective.

Chromosome individual 1					
Variable:	x_p	$x_{p,w}$	$x_{p,e}$	$o_{p,p'}$	$x_{p,e,w}$
σ	0	1	0	1	1
ρ	0.5	0.4	0.6	0.7	0.3
Chromosome individual 2					
Variable:	x_p	$x_{p,w}$	$x_{p,e}$	$o_{p,p'}$	$x_{p,e,w}$
σ	0	0	0	1	0
ρ	0.7	0.3	0.4	0.3	0.8
Chromosome individual after crossover					
Variable:	x_p	$x_{p,w}$	$x_{p,e}$	$o_{p,p'}$	$x_{p,e,w}$
σ	0	1	0	1	0
ρ	0.6	0.4	0.6	0.5	0.6

Table 3.1: An example of two chromosome individuals

The encoding technique used for evolutionary optimization is based on the approach proposed in [35]. We hereby provide a brief overview of the same, and for more details please refer to [35]. Each variable is assigned a priority value $\rho, \rho \in \mathbf{R}^+$ and a binary value $\sigma, \sigma \in 0, 1$. The value of σ defines the the decision phase of each variable while the value of ρ defines the priority of the variable in the searching process, i.e., the variable with higher priority will be handled first. Two example chromosome individuals are shown in Tab. 3.1. In the evolutionary optimization presented here, the crossover on the value of σ is a simple selection between individuals while the crossover on the real value ρ is performed by the simulated binary crossover (SBX) [36]. Searching in the solution space is based on backtracking technique which tracks back when the assigned value is infeasible for a variable.

The evolution through the generations is based on the objective values and the quality of the solutions improves with the iteratively reproduction and selection. The inferior solutions are removed to ensure a convergence toward an optimal value. In each generation, a Pseudo Boolean (PB) solver is used to find the feasible solutions

that meet all the linear constraints for each chromosome.

3.4.9 Performance Evaluation

This section evaluates the proposed parallel transmission algorithms with different buffer settings. The numerical results are two-fold. First, the optimal parallel transmission algorithm and EA-based algorithm are evaluated and compared in small-scale networks considering the complexity issue of the ILP optimization. After demonstrating the capability of EA-algorithms to compute near-optimal solutions, we proceed to evaluate the performance of the parallel transmission in a large network topology shown in Fig. 3.9. The comparison between the ILP optimization and EA-based algorithm utilizes a single objective function defined in Eq. (3.5). When only the EA-based algorithm is evaluated, both objective functions defined in Eq. (3.5) and Eq. (3.6) are used. The multi-objective optimization aims to provide an optimal solution with minimum resource consumption per connection demand while minimizing the differential delay between paths. All the connection demands arrive in a Poisson process and are uniformly distributed among all node pairs.

The number of wavelengths per fiber link is scaled down to 16 in order to reduce the run-time in the simulation. The number of wavelengths required by the connection demands is also scaled down accordingly. The maximum number of wavelengths required by the connection demands is assumed to be five. The minimum bandwidth requirement is assumed to be one wavelength. The number of connection demands arriving at the OTN/WDM network is in inverse proportion to their bandwidth requirement, i.e., $1\lambda : 2\lambda : 3\lambda : 4\lambda : 5\lambda = 5 : 4 : 3 : 2 : 1$. Nevertheless, the simulation scenario in our study aligns with guidance of the proposals in IEEE 802.3ba, which have suggested that both 40 Gbps and 100 Gbps Ethernet can be supported by 4λ, e.g., 4×10 Gbps for 40 Gbps Ethernet and

Algorithm 4: Evolutionary optimization for parallel transmission

1 **Input:** $G(V, E, W_e)$; Connection request $R(s, d, r)$; The parallel transmission model

2 **Ouput:** optimal set of paths for R

Step 1 *Map variables (x_p,$x_{p,w}$, $x_{p,e}$, $o_{p,p'}$, $x_{p,e,w}$) from the optimization model into chromosome space*

- Encode the binary variables into offspring in chromosome space.
- Generate N offspring in one generation by using the evolutionary algorithm.

Step 2 *Find feasible solutions for each generation*

for *each offspring in one generation* **do**

Use encoded variables as input into a Pseudo Boolean (PB) solver [37].

Find feasible solutions which fulfill all linear constraints.

end

Step 3 *Selection of the solutions with multi-objective optimization*

- Calculate the objective values defined in Eq.(3.5) and Eq.(3.6) of all solutions and and evaluate the optimality of the remaining solutions. The linear constraints are used to eliminate the unfeasible solutions.
- Sort the solutions according to the optimality and remove m inferior solutions from the solution space.

Step 4 *Evolution* Generate m new offspring by the evolutionary operations and go back to step 2.

Step 5 *Stop and output an optimal solution after the given number of generations' evolution*

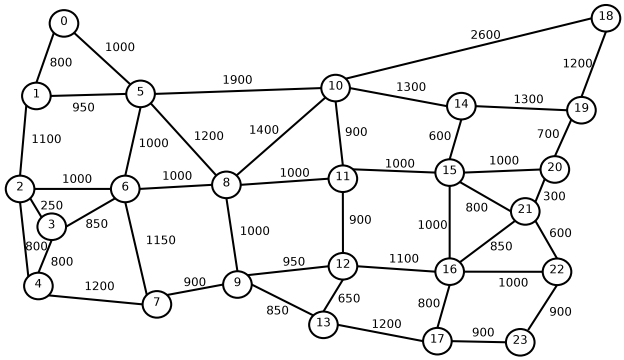


Figure 3.9: The USA national network topology [1]

4×25 Gbps for 100 Gbps Ethernet [2]. The value of $|P|$ implies the maximum number of diverse routes can be used. For instance, $|P| = 2$ means that wavelengths can be assigned over two physical paths (fiber-level paths) for a connection demand. When $|P| = 1$, it means that all the wavelengths are assigned over the same fibers.

The *load* A (in Erlang (Erl)) used in in this section is defined as $u * h * r/C$, where u is connection arrival rate and h is the mean connection holding time. r and C represent the average bandwidth requirement and the capacity of one wavelength respectively. *Bandwidth Blocking Ratio* is used as a performance metric, which is defined the percentage of the requested bandwidth of blocked connections in the total requested bandwidth. This is due to the fact that different connection demands request different numbers of wavelengths, whereby the resulting blocking has larger impact on the connection demands with larger bandwidth requirements.

The optimization is invoked for 2500 requests at each network load to derive a meaningful average value. The number of evolution generations is set to 150 with 25 individual offspring in each generation, which leads to 3750 evaluations of the objective functions per re-

quest. For each results set, the values are averaged over 30 runs. All the results presented are obtained with the implementation based on the open source optimization framework OPT4J [37].

Comparison of Optimization and EA-based Algorithms

The comparison of the ILP optimization and EA-based algorithm is based on randomly generated small networks. The number of nodes in the networks, denoted as $|V|$ increases from 2 to 15. $|E| = 4{\cdot}|V|-6$ links are randomly added into each graph. Both ILP model and EA-based algorithms are evaluated in each network instance with number of fiber-level paths varying from one to three, i.e., $|P| = 1, 2, 3$. The bandwidth requirements are generated following the distribution as mentioned above. Each fiber link has 16 wavelengths and it is assumed that differential delay is compensated by electronic buffer. Both ILP and EA-based algorithm are run 30 times to derive an average value. For fairness in the comparison, both algorithms use the same objective function (Eq. 3.5), while available electronic buffer is used as the constraint.

The evaluation metrics used in this study are defined as follows:

- *Scalability* is defined as the average runtime to obtain an optimal solution.
- *Optimality* is referred to the quality of a solution, which is defined in terms of the normalized hypervolume [38].

The quality of a solution is defined in terms of the normalized hypervolume [38] which utilizes hypervolume of the optimal solution (or best solution) as the base. Quality of a solution A is thus defined as $\frac{Hypervolume(A)}{Hypervolume(Optimal/Best)}$. Fig. 3.10 depicts that the evolutionary optimization can yield solutions with same quality as ILP optimization within a reasonable amount of time. When ILP optimization fails to find an optimal solution, the best solution obtained in evolutionary optimization is used as the normalization base.

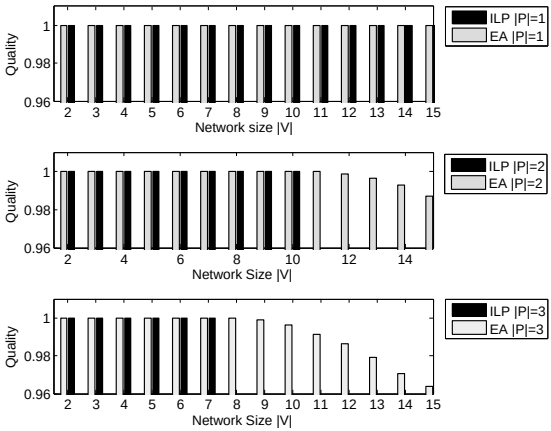


Figure 3.10: The quality of solutions comparison

Fig. 3.11 shows the scalability of ILP and EA-based algorithm in different network settings with y-axis in log-scale. It is shown that the ILP performs well when the number of fiber-level paths is restrict to one, i.e. $|P| = 1$. However, the average run-time increases rapidly when $|P| > 1$. Fig. 3.11 also shows that the ILP optimization can not find a solution with large networks, i.e., at most $|V| = 10$ with $|P| = 2$ and $|V| = 7$ with $|P| = 3$, respectively. In contrast, the EA-based algorithm shows a good scalability by obtaining solutions within a few seconds, regardless of the increasing network size. The scalability study implies that ILP optimization can not obtain optimal solutions for parallel transmission in a realistic scale network due to the complexity issue, either caused by the increasing network size or the number of paths.

Evaluation with Electronic Buffers

As previously shown that the ILP optimization is restricted by the complexity issue even in small networks. Hence, only EA-based al-

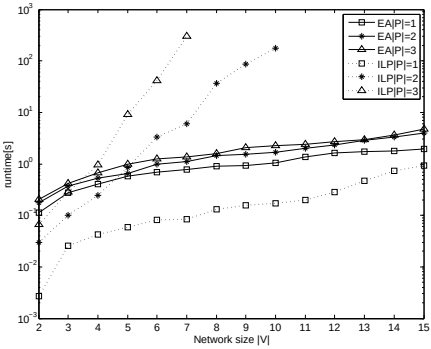


Figure 3.11: The scalability comparison

gorithm is evaluated in a large network shown in Fig. 3.9. The simulation scenario with regard to the number of wavelength per fiber, request arrival etc. are the same as described above. The available electronic buffer at each node is randomly assigned with a value between 5 MB and 10 MB.

Bandwidth Blocking Ratio: Fig. 3.12 shows the bandwidth blocking ratio with network load ranging from $50Erl$ to $100Erl$. It shows that allowing multiple wavelengths allocated over multiple physical paths (fiber-level paths), i.e. $|P| > 1$, can decrease the bandwidth blocking ratio. However, spreading traffic over more paths does not necessarily lead to the improvement of network performance regarding the bandwidth blocking ratio. As shown in Fig. 3.12, slight improvement has been observed with $|P| = 4$ comparing with $|P| = 3$, while $|P| = 5$ has almost identical performance as $|P| = 4$. We further study the relation between the number of paths and the bandwidth requirement and show results at a high network load ($100Erl$) in Fig. 3.13. The advantage of parallel transmission over diverse paths in high network load is observed especially with the increasing number of requested wavelengths. However, the same phenomena has been observed, that is, $|P| = 4$ and $|P| = 5$ do not improve

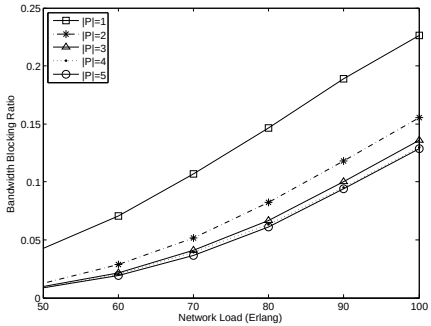


Figure 3.12: The impact of $|P|$ on bandwidth blocking ratio vs. network load.

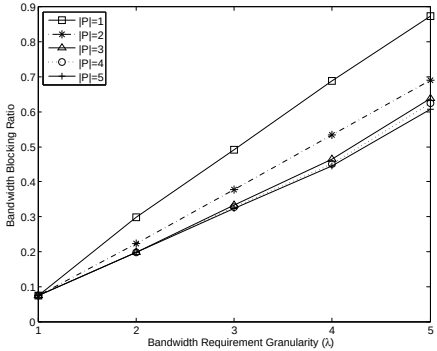


Figure 3.13: Bandwidth blocking ratio vs. requested bandwidth; A=100 Erl.

the performance significantly comparing with $|P| = 3$ in terms of the bandwidth blocking ratio. In the following, we will only show results with $|P|$ up to 3.

Maximal Differential Delay and Required Buffer: While the parallel transmission over diverse paths in optical networks can decrease the bandwidth blocking ratio, it may lead to the cost of elec-

tronic buffering. The increasing number of paths used for a single connection leads to an increase of differential delay, which opens up a possibility of overflowing the electronic buffer at the destination. We therefore evaluate the resulting differential delay and buffer requirement of the solutions yielded from our optimization model. Fig. 3.14 shows the average buffer requirements of the solutions. A larger $|P|$ requires a larger buffer due to the need to perform re-ordering. However, all the solutions found require at most 3.3MB on average, which is lower than the minimum available buffer size in the network, i.e., 5MB. The average differential delay is increased with the increase of $|P|$ as shown in Fig. 3.15.

To better illustrate the differential delay of the optimal solutions obtained from the presented optimization model, we take $|P| = 2$ as an example and show the maximal differential delay and the average differential delay of the computed paths in Fig. 3.16. It can be seen that the value of the maximum differential delay increases with the network load. However, a maximal value of $3000us$ has been observed in all the solutions with $|P| = 2$, requiring 3.7MB buffer (refer to Fig. 3.14), which is smaller than the available buffer size.

Evaluation with Optical Buffers

To assess the impact of using FDL-based optical buffers, we first study the differential delay and buffer requirement of the solutions computed by the optimization model. Fig. 3.14 shows the average buffer requirements of the solutions. A larger $|P|$ requires a larger buffer due to the need of frame re-ordering. However, all the solutions found require at most 3.3MB on average, which is within the bound of available buffer size in the network, i.e., 5MB. The average differential delay increases with increasing $|P|$ as shown in Fig. 3.15.

Fig. 3.15 and Fig. 3.16 depict that the differential delay of the parallel transmission over diverse paths in optical layer is decreased

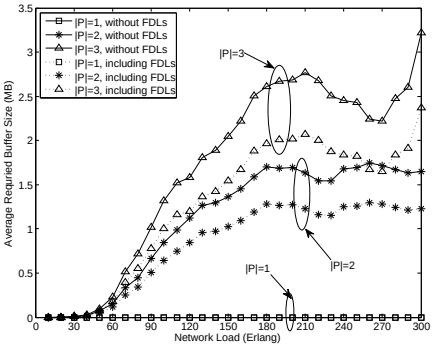


Figure 3.14: Average buffer size vs. network load without and with FDLs

with optimally utilizing FDLs, which leads to a smaller electronic buffering requirement. Fig. 3.14 shows that the average buffer size required is decreased by 30% for $|P| = 3$ and 20% for $|P| = 2$ at the high network loads (here, from $150Erl$ to 300 Erl) by using FDLs. However, including FDLs only slightly reduces the bandwidth blocking ratio as shown in Fig. 3.17. This is because the FDLs can only change the path delay in discrete time units, which has limited impact on blocking probability.

Evaluation of Bufferless Parallel Transmission

Bandwidth blocking ratio: We first study the impact of frame size on the bandwidth blocking ratio. Due to the fact that the same trend has been observed with different $|P|$, we therefore show a set of representative results with $|P| = 2$. The frame sizes used are 5Mb, 10Mb and 50Mb and the data rate of an optical channel is assumed to be 10Gbps. As illustrated in Fig. 3.18, frame size that decided in the Ethernet layer does not affect the bandwidth blocking

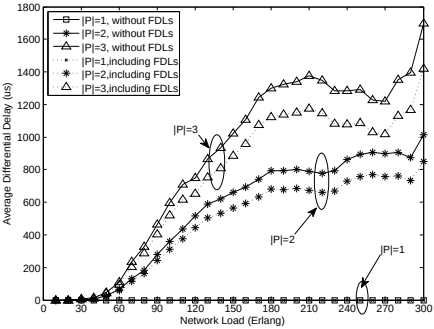


Figure 3.15: Average differential delay of various $|P|$ with and without FDLs

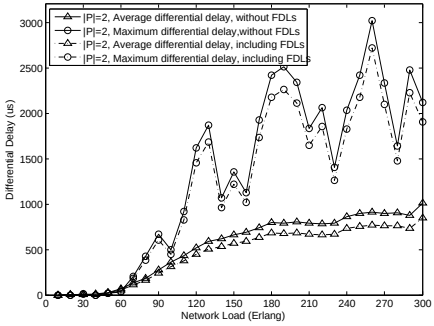


Figure 3.16: Max. differential delay with $|P| = 2$ without and with FDLs

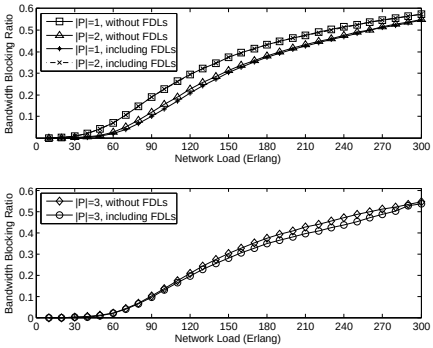


Figure 3.17: The impact of FDLs on bandwidth blocking ratio

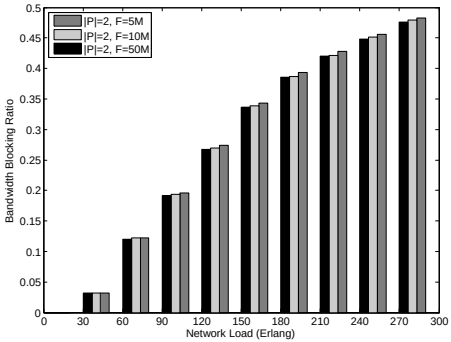


Figure 3.18: Frame size vs. bandwidth blocking ratio vs. network load; $|P| = 2$

ratio in the OTN/WDM network. Of note is the performance with small frame size, i.e., F= 5Mb, which has a comparable performance of bandwidth blocking ratio as big frame sizes. This set of results implies that parallel transmission in OTN/WDM networks is feasible even with diverse paths and it is independent from the frame size determined in the Ethernet layer.

Impact of frame size and $|P|$ on differential delay: We also

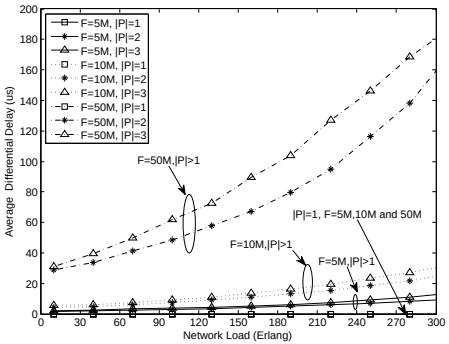


Figure 3.19: Average differential delay of bufferless parallel transmission

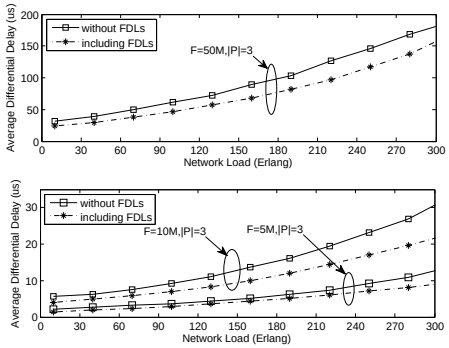


Figure 3.20: Impact of FDLs on average differential delay in bufferless parallel transmission vs. network load

study the impact of frame size and the size of $|P|$ on the average differential delay of the solutions from our optimization model for parallel transmission. As shown in Fig. 3.19, the first observation is that the increasing number of paths used for a connection demand increases the average differential delay. For example, $|P| = 3$ always results in a larger average differential delay compared to $|P| = 2$ with the same frame size. The second observation is that frame size can

affect the average differential delay, despite it does not affect the bandwidth blocking ratio. Take $|P| = 2$ as an example, F=50Mb results in much higher differential delay in comparison to F=10Mb and F=5Mb. The difference is as high as $70\mu s$ between F=50Mb and F=10Mb. While the difference is relatively small between F=10Mb and F=5Mb, the differential delay of F=5Mb is still about 50% less than F=10Mb. In all cases, the average differential delay is lower than $200\mu s$, which is small considering the size of network under study.

Impact of available FDLs on bufferless parallel transmission: Finally, we show that the bufferless parallel transmission mechanism can also benefit from the available FDLs to reduce the average differential delay. Due the similar trend observed in the study, we only show a set of representative results with $|P| = 3$ in Fig. 3.20. To have a closer look at the impact of FDLs on the bufferless parallel transmission with small frames, we show results of F=5Mb and F=10Mb in a separate diagram from F=50Mb. As shown in Fig. 3.20, usage of FDLs can decrease the average differential delay by $10\mu s$ to $50\mu s$ for F=50Mb. About $5\mu s$ to $10\mu s$ reduction can be achieved for F=10Mb in case network load is high ($\geq 100Erl$), while slight improvement can be obtained in case of F=5Mb, which is about $5\mu s$ at most.

3.5 Summary

This chapter presented the first study on feasibility of parallel transmission in OTN/WDM networks to support high-speed Ethernet. We proposed a novel architecture which integrates the parallelism in Ethernet layer and and parallel transmission in OTN/WDM networks. The proposed architecture is in compliance with standards of both Ethernet and optical layers, i.e., IEEE 802.3ba and ITU-T G.709. We modeled a MRWA problem tailored for parallel transmis-

sion in OTN/WDM networks, which has not been studied to date.

We proposed an ILP based optimization model with consideration of both electronic buffer and FDL based optical buffer. To counter the complexity issue of ILP optimization, we utilized an EA based optimization approach to find optimal or near-optimal solutions. We also demonstrated that FDLs can alleviate the requirement of electronic buffers in parallel transmission. The technological restrictions imposed by the discrete delay units of FDLs can be overcome by optimally designed parallel routing. We also showed that it is feasible to apply parallel transmission in optical networks without requiring buffers with the proposed bufferless parallel transmission method.

4

Parallel Transmission in Flexi-grid Optical Networks

4.1 Introduction

Elastic optical network based on Orthogonal Frequency Division Multiplexing (OFDM) technology is a valid parallel transmission solution in the optical layer, due to its fundamental parallel nature. With OFDM, optical spectrum in flexi-grid optical networks is *sliced or parallelized* into a sequence of frequency slots and signals are modulated on frequency slots in form of sub-carriers, with each sub-carrier at a lower data rate [39]. Since the sub-carriers are orthogonal to each other in the frequency domain, signals modulated on them can be received in parallel without interference.

This chapter investigates feasibility of using OFDM-based optical networks to support high-speed Ethernet, based on the works [7] [40] [41] that were the first to address the same issue. The modeling and analysis in this chapter focus on two main issues: 1) the suitability to support high-speed Ethernet; 2) the capability of reducing *spectrum fragmentation*.

We propose a novel architecture for high-speed Ethernet transmission over an OFDM-based optical network, which can flexibly assign parallel sub-carriers for traffic from parallel Ethernet lanes. The proposed architecture is in full compliance with IEEE and ITU-T standards, hence can be practically deployed. Similar as Routing

and Wavelength Assignment (RWA) algorithms in WDM networks, Routing and Spectrum Allocation (RSA) algorithms are used in elastic optical networks to set up connections. Until now, studies on RSA have focused on single path routing, where spectrum fragmentation remains an open issue.

In this chapter, we formulate an optimization model based on ILP and propose heuristic algorithms for dynamic computation of multiple parallel paths and spectrum assignment, referred to as Multipath Routing and Spectrum Allocation (MRSA) problem. Both uniform modulation format and distance-adaptive modulation format assignment are studied.

4.2 Supporting Publications

1. X. Chen, A. Jukan, A. Gumaste, "Optimized Parallel Transmission in Elastic Optical Networks to Support High-Speed Ethernet," IEEE/OSA Journal of Lightwave Technology(JLT), Volume 32, Issue 2, January 2014 PP. 228-238.

2. X. Chen, Y. Zhong, A. Jukan, "Multipath Routing in Elastic Optical Networks with Distance-adaptive Modulation Formats," in IEEE International Conference on Communications (ICC), June 2013.

3. X. Chen, A. Jukan, A. Gumaste, " Multipath De-fragmentation: Achieving Better Spectral Efficiency in Elastic Optical Path Networks," in IEEE International Conference on Computer Communications (INFOCOM), April 2013, PP. 390-394.

4.3 Preliminary

In elastic optical networks, the spectrum is sliced into frequency slots with granularity finer than that is currently used in WDM networks,

i.e., 50GHz or 100GHz. As illustrated in Figure 4.1 [6], the frequency slots are based on the ITU-T fixed grids, i.e., the central frequency is allocated at 193.1THz. The width of a frequency slot depends on the specific transmission systems. In this example, one frequency slot is 12.5GHz. In OFDM-based elastic optical networks, traffic is carried by sub-carriers and each sub-carrier is modulated on a frequency slot. Upon receiving a connection request, a group of frequency slots, usually consecutive in frequency domain, are assigned accordingly.

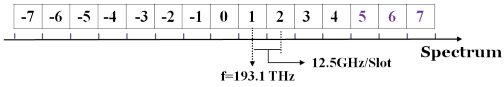


Figure 4.1: Frequency slot approach to slice spectrum [6]

4.4 Reference Architecture

Fig. 4.2 shows the proposed reference architecture for parallel transmission in an OFDM-based elastic optical networks to support 100GE [7]. The OTN layer is an adaptation layer between Ethernet layer and optical transmission layer. Ethernet traffic is mapped onto Optical channel Data Unit (ODU) first in OTN layer before it is modulated onto sub-carriers in optical OFDM networks. Currently, IEEE 802.3ba specifies two schemes for 100GE, i.e., 4×25 Gbps and 10×10 Gbps. Without loss of generality, an asymmetric mapping between Ethernet lanes and sub-carriers is shown as an example. 100GE signal is distributed into 4×25 Gbps lanes via Multi-lane distribution interface. OTN layer decides the size of ODU based on the capacity of the sub-carriers.

Assume each sub-carrier can support an ODU channel (10 Gbps), traffic from an Ethernet lane at 25 Gbps can not be directly modulated onto a sub-carrier. An ODU4 channel is therefore used first,

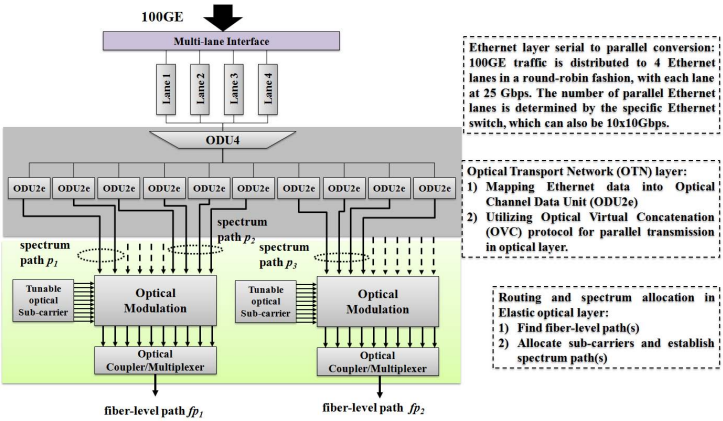


Figure 4.2: Reference architecture for parallel transmission in optical OFDM networks to support high-speed Ethernet [7]

which is inverse-multiplexed into 10 ODU2e channels. After that, each ODU2e is mapped into Optical channel Transport Unit (OTU2e) and modulated on a sub-carrier utilize a frequency slot. In the example shown in Fig. 4.2, 100GE traffic is modulated onto 10 sub-carriers (dashed lines are the subcarriers already "allocated").

In OFDM based elastic optical networks, an end-to-end channel is referred to as a spectrum path which is allocated with a group of consecutive frequency slots [42]. Given the resource availability in the example shown in Fig. 4.2, the 100GE connection is supported by establishing three spectrum paths, i.e., p_1, p_2 and p_3. As it is illustrated, the spectrum paths can traverse different fibers, referred to as fiber-level paths. In Fig. 4.2, two fiber-level paths are used for this 100GE connection, i.e., fp_1 and fp_2.

4.5 Challenges and Issues

Differential delay issue: In OFDM-based optical networks, number of frequency slots assigned to a spectrum path can vary in a wide range. Hence, two types of differential delay are considered, namely, fiber effects caused differential delay and path diversity cased differential delay. Fiber effects caused differential delay is mainly due to the Group Velocity Dispersion (GVD) effect in the fibers. Sub-carriers on different frequencies travel (in form of waves) at different speeds. An approximation of the maximum delay difference caused by GVD in a spectrum band is as follows:

$$\Delta d_{max} \approx D(f_c) \cdot (f_{max} - f_{min}) \cdot L \tag{4.1}$$

where $D(f_c)$ is the fiber dispersion at the central frequency; f_{max} and f_{min} are the highest frequency and smallest frequency of the spectrum band, see [43]; and L is the transmission distance. The condition of Eq. (4.1) is that the central frequency is much larger than $(f_{max} - f_{min})$. The OFDM-based optical networks follow the same spectrum dimension as it is in *"fix grid"* [39], i.e., the central frequency is $f_c = 193.1$ THz, which is much larger than the frequency difference in any spectrum band. Hence, Eq. (4.1) can be directly applied. As an example, we assume that 100 GE utilizes a spectrum band composed of 10 consecutive sub-carriers for parallel transmission and also assume that each frequency slot is 50 GHz[1], i.e., channel spacing is 0.4nm. The fiber dispersion of a Single Mode Fiber (SMF) is 17 $ps/nm/km$ at the central frequency [43]. Hence, the maximum differential delay caused by dispersion in the parallel transmission is $\Delta d_{max} \approx 0.68$ μs for a connection with physical distance of 1×10^4 km.

Another, and more commonly considered, type of differential delay in parallel transmission is caused by the path diversity. When

[1] A frequency slot is generally smaller than 50 GHz. However, we take the standardized channel spacing, i.e., 50 GHz as an example here.

spectrum paths traverse different physical paths (fiber-level paths), the different transmission distance along different paths also leads to the differential delay. The differential delay caused by the path diversity can be simply calculated as L/v, where v is the signal propagation speed in the path and L is the path length. Considering the standard SMF where the signal propagation speed is 2×10^5 km/s, for a connection with length of 1×10^4 km, the maximum differential delay between spectrum bands would be 50 ms.

Spectrum fragmentation issue: The nature of connection demands can be fine and coarse grained, which leads to the so-called *spectrum fragmentation issue* in elastic optical networks. High-speed Ethernet may exacerbate this problem due to the high-bandwidth requirements. So far, Routing and Spectrum Assignment (RSA) algorithms in the literature focused on finding a single spectrum path and allocating consecutive frequency slots. With current technical constraints in elastic optical networks, a spectrum path requires same frequency slots continuously from source to the destination. Establishing such connections for high-bandwidth flows can easily fragment the spectrum on optical links, leading to the rejection of the future connections.

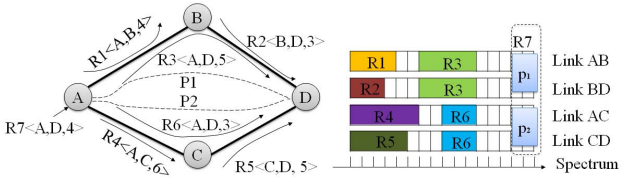


Figure 4.3: An illustrative example of spectrum fragmentation and a solution with two non-consecutive spectrum slices

To better understand this problem, Fig. 4.3 illustrates an example, where optical spectrum on each fiber link is assumed to be sliced into 16 frequency slots, with one sub-carrier on each frequency slot.

The traffic demand is defined as $R(S, D, T_r)$, where S, D and T_r denote source, destination, and the number of required sub-carriers, respectively. Assuming that 2 sub-carriers are assigned as the guard-band (typically used to insulate the adjacent spectrum paths), the spectrum on all fiber links are fragmented after the allocation of 6 spectrum bands, i.e., $R_1 - R_6$ in this example.

Upon the arrival of R_7, the request is rejected when the network tries to provide a single spectrum path, even though there are sufficient sub-carriers available in the network. This phenomena is particularly pronounced in case of high-speed Ethernet services with a transmission rate of 40/100 Gbps which require a large number of consecutive sub-carriers. However, this issue can be effectively resolved by parallel transmission over multiple spectrum paths. In the example shown in Fig. 4.3, R_7 can be accommodated by two spectrum paths, i.e., p_1 along links AB-BD and p_2 along links AC-CD, with two frequency slots per spectrum path.

4.6 Summary of Notations and Terminology

Routing and Spectrum Allocation (RSA) problem in elastic optical networks has its counterpart in WDM networks, i.e., Routing and Wavelength Assignment (RWA) problem. To avoid confusion with RWA problem and ease the understanding of this chapter, we first clarify the terminologies as follows:

- *Sub-carrier* is a channel which carries signals in optical OFDM networks. Sub-carriers are orthogonal to each other. A sub-carrier utilizes one frequency slot with size of s_f .

- *Guard-band (GB)* is a slice of spectrum (expressed in sub-carriers) used to insulate two adjacent spectrum paths.

- *Spectrum path*, denoted as p, is a spectrum slice allocated continuously from source to destination in form of a group of con-

secutive sub-carriers.

- *Fiber-level path*, denoted as fp, is a physical route from source to destination, over which one or multiple spectrum paths can be established.

Take the example shown in Fig. 4.2, p_1, p_2 and p_3 are three spectrum paths and fp_1 and fp_2 are two fiber-level paths. The fiber-level path fp_1 contains two spectrum paths, p_1 and p_2, while fp_2 contains only one spectrum path p_3. Notions used in this chapter are summarized in Table 4.1.

Parameter	Description
$G(V, E)$	A graph represents an elastic optical network with nodes in set V and edges in set E
$R(S, D, T_r)$	Connection request; S, D and T_r denote source, destination, and the number of required sub-carriers, respectively.
f_i	A sub-carrier with index i
s_f	The size of a frequency slot
F	An ordered set contains all frequency slots (sub-carriers) on a link, $F = \{f_1, f_2, ..., f_N\}$
F^e	A set contains available frequency slots on link e
LD_e	Delay of the link e
L_e	Length of link e
GB	Guard-band
M	Maximum acceptable differential delay
p	Spectrum path
fp	Fiber-level path
$\mathcal{P}$	Set of all the spectrum paths computed for R
$\mathcal{FP}$	Set of fiber-level paths computed for R
K	Maximum number of fiber-level paths can be used, $\lvert\mathcal{FP}\rvert \leq K$

Table 4.1: Notations

4.7 Uniform Modulation Format Assignment

This section presents the Multipath Routing and Spectrum Allocation (MRSA) algorithm with assumption that all sub-carriers have the same transmission rate, i.e., all sub-carriers have uniform modulation format. This assumption simplifies the mapping between OTN frames and sub-carriers, thus facilitating the actual implementation of optical parallel transmission in practice. We also assume that there are sufficient transponders to support the spectrum paths. This assumption can be relaxed by constraining the overall number of spectrum paths not to exceed the available transponders.

The analysis of differential delay issue shown earlier has indicated that the main factor of differential delay in parallel transmission is path diversity. Considering the same transmission distance, the maximum differential delay caused by the GVD among sub-carriers is insignificant comparing with the delay difference caused by propagation, e.g., 0.68 μs vs. 50 ms. Therefore, evaluation of the algorithm proposed in this section will only consider the differential delay caused by path diversity. However, the optimization model still accounts for the issue of differential delay among sub-carriers, which maybe useful for new materials inducing new fiber propagation properties. Finally, we assume that the differential delay is compensated in the OTN layer in our architecture.

4.7.1 ILP Optimization Model

The proposed ILP optimization model [7] relies on the variables defined in Table 4.2. The objective of the ILP model is defined as minimizing the total number of sub-carriers allocated to a connection request, i.e.,

$$Minimize \sum_{p \in \mathcal{P}} \sum_{f_i \in F} \sum_{e \in E} x_{p,e,i} \tag{4.2}$$

Variable	Description
x_p	Binary variable; it equals to 1 if a spectrum path p is found for R; otherwise it is 0
$x_{p,e}$	Binary variable; it equals to 1 if a spectrum path p uses e, otherwise it equals to 0
$y_{p,i}$	Binary variable; it equals to 1 if a spectrum path p uses sub-carrier $f_i \in F$, otherwise it equals to 0
$x_{p,e,i}$	Binary variable; it is 1 if a spectrum path p uses sub-carrier $f_i \in F$ on link $e \in E$, otherwise it is 0
$o_{p,p'}$	Binary variable; it equals to 1 if spectrum paths p and p' share at least one link, otherwise it is 0
pd_p	Integer variable; it denotes delay of spectrum path p
T_p	Integer variable; it denotes the number of sub-carriers allocated to the spectrum path p
GVD_p	Integer variable; it denotes the differential delay caused by GVD on the spectrum path p

Table 4.2: Variables

subject to the constraints defined as follows:

***Routing constraints*:** Eq. (4.3) ensures that traffic on the routed path can be added and dropped only at source and destination nodes, respectively. Constraint defined in Eq. (4.4) guarantees that a spectrum path starts from the source node and ends at the destination node. Finally, Eq. (4.5) eliminates the loops at source and destination nodes.

$$\forall p \in \mathcal{P}, \tilde{v}, v \in V, v \neq s, d : \sum_{e=(\tilde{v},v)\in E} x_{p,e} = \sum_{e=(v,\tilde{v})\in E} x_{p,e} \tag{4.3}$$

$$\forall p \in \mathcal{P}, \tilde{v} \in V, \tilde{v} \neq s, d : \sum_{e=(\tilde{v},d)\in E} x_{p,e} = \sum_{e=(s,\tilde{v})\in E} x_{p,e} = x_p \tag{4.4}$$

$$\forall p \in \mathcal{P}, \tilde{v} \in V, \tilde{v} \neq s, d : \sum_{e=(\tilde{v},s)\in E} x_{p,e} = \sum_{e=(d,\tilde{v})} x_{p,e} = 0 \tag{4.5}$$

***Spectrum continuity constraint*:** We restrict that all spectrum paths to be all-optical between source and destination nodes, i.e.,

restricted by spectrum continuity constraint. Eq. (4.6) indicates that sub-carrier with index i is assigned to the spectrum path p from the source node. Eq. (4.7) specifies that a spectrum path can only use sub-carriers with same index on all fiber links it traverses.

$$\forall p \in \mathcal{P}, \tilde{v} \in V, \tilde{v} \neq s : y_{p,i} = \sum_{e=(s,\tilde{v}) \in E} x_{p,e,i} \tag{4.6}$$

$\forall f_i \in F, p \in \mathcal{P}, \tilde{v}, v \in V \setminus \{s, d\}$:

$$\sum_{e=(\tilde{v},v) \in E} x_{p,e,i} = \sum_{e=(v,\tilde{v}) \in E} x_{p,e,i} \tag{4.7}$$

***Spectrum consecutive constraints*:** For efficient modulation, consecutive sub-carriers are required in a spectrum band when it is assigned to a spectrum path [39]. The spectrum consecutive constraints are defined in Eq. (4.8) and Eq. (4.9). Eq. (4.8) determines the number of sub-carriers allocated to path p. When two sub-carriers with index f_i and f_j $(j \geq i)$ are used for p, the right-hand side of Eq. (4.9) equals to T_p. This constraint ensures that the gap between two sub-carriers should be equal to or less than T_p. When f_i and f_j are not used at the same time, the right-hand side of Eq. (4.9) results in an infinite value, which keeps Eq. (4.9) true.

$$\forall p \in \mathcal{P}, e \in E, v \in V, f_i \in F : T_p = \sum_{e=(s,v)} \sum_{i} x_{p,e,i} \tag{4.8}$$

$\forall f_i, f_j \in F, j \geq i, p \in \mathcal{P}, e \in E$:

$$f_j \cdot x_{p,e,j} - f_i \cdot x_{p,e,i} + 1 \leq Tp + (2 - x_{p,e,i} - x_{p,e,j}) \cdot \infty \tag{4.9}$$

***Non-overlapping constraints*:** To avoid collision, a sub-carrier can not be assigned to multiple spectrum paths simultaneously. The binary variable $o_{p,p'}$ is defined to denote if two spectrum paths p and p' have at least one common link. The value of $o_{p,p'}$ is determined by Eq. (4.10) and Eq. (4.11). When path p and p' share at least one fiber link, $o_{p,p'}$ equals to 1. Otherwise, $o_{p,p'}$ equals to 0. Eq. (4.12)

specifies that a spectrum slot f_i can not be assigned to p and p' at the same time if two spectrum paths have common links, i.e., either $y_{p,i}$ or $y_{p',i}$ can be equal to 1 when $o_{p,p'} = 1$. Finally, Eq. (4.13) defines that spectrum assignment only happens when a spectrum path p is used by the connection.

$$\forall p, p' \in \mathcal{P}, p \neq p', e \in E : x_{p,e} + x_{p',e} - o_{p,p'} \leq 1 \tag{4.10}$$

$$\forall p, p' \in \mathcal{P}, e \in E : o_{p,p'} \leq \sum_{e} x_{p,e} \cdot x_{p',e} \tag{4.11}$$

$$\forall p, p' \in \mathcal{P}, p \neq p', f_i \in F : y_{p,i} + y_{p',i} + o_{p,p'} \leq 2 \tag{4.12}$$

$$\forall p \in P, f_i \in F : x_p - y_{p,i} \geq 0 \tag{4.13}$$

The constraint defined in Eq. (4.11) is non-linear. We define a new binary variable denoted as $\gamma_{p,p',e} = x_{p,e} \cdot x_{p',e}$, and linearize Eq. (4.11) as follows:

$$\forall p, p' \in \mathcal{P}, e \in E : o_{p,p'} \leq \sum_{e} \gamma_{p,p',e} \tag{4.14}$$

$$\forall p, p' \in \mathcal{P}, e \in E : \gamma_{p,p',e} \leq x_{p,e} \tag{4.15}$$

$$\forall p, p' \in \mathcal{P}, e \in E : \gamma_{p,p',e} \leq x_{p',e} \tag{4.16}$$

$$p, p' \in \mathcal{P}, e \in E : x_{p',e} + x_{p,e} - \gamma_{p,p',e} \leq 1 \tag{4.17}$$

***Guard-band constraint*:** The constraint defined in Eq. (4.18) specifies that the spectrum assignment only happens when the available sub-carriers are sufficient to meet the guard-band requirement. When a sub-carrier f_i is allocated to a path p, a sub-carrier within the range $\{f_i - GB, f_i + GB\}$ cannot be allocated to other paths, i.e., all sub-carriers within the range $\{f_i - GB, f_i + GB\}$ are excluded from the available spectrum set of link e for other paths. Eq. (4.19) ensures that a guard-band exists between two spectrum paths p and p', if they share at least one common link. When p and p' have no common link, $o_{p,p'}$ equals to 0, which guarantees that Eq. (4.19) is true.

$$\forall p \in \mathcal{P}, e \in E, \{f_i \pm GB\} \in F \setminus F^e : x_{p,e,i} = 0 \tag{4.18}$$

$$\forall p, p' \in \mathcal{P}, e \in E, f_i, f_j \in F : |f_j \cdot x_{p,e,j} - f_i \cdot x_{p',e,i}| \geq GB \cdot o_{p,p'} \tag{4.19}$$

***Bandwidth constraint*:** The constraint defined in Eq. (4.20) ensures that the number of sub-carriers assigned to all the spectrum paths for connection request R is equal to the traffic demand T_r.

$$\sum_{p \in \mathcal{P}, f_i \in F} y_{p,i} = T_r \tag{4.20}$$

***Differential delay constraint*:** Differential delay in optical parallel transmission is caused by the fiber effects and path diversity, as it is discussed in Sec.II B. The maximum differential delay caused by the GVD on path p is calculated based on the Eq. (4.1). As shown in Eq. (4.21), $T_p \cdot s_f$ is the gap between the highest frequency and the lowest frequency of a spectrum band. $\sum_e L_e \cdot x_{p,e}$ calculates the length of the path p.

$$\forall p \in \mathcal{P}, e \in E : GVD_p = D(f_c) \cdot s_f \cdot T_p \cdot \sum_e L_e \cdot x_{p,e} \tag{4.21}$$

Note that Eq. (4.21) is non-linear. We hence define an integer variable, denoted as $z_{p,e} = T_p \cdot x_{p,e}$. The GVD constraint can be linearized as shown in Eq. (4.22).

$$\forall p \in \mathcal{P}, e \in E : GVD_p = D(f_c) \cdot s_f \cdot \sum_e L_e \cdot z_{p,e} \tag{4.22}$$

The variable $z_{p,e}$ is restricted by the constraints defined in Eq. (4.23), Eq. (4.24) and Eq. (4.25), which determine the value of $z_{p,e}$ to be either zero or equal to T_p. The delay of p is defined in Eq. (4.26). The total differential delay is calculated in Eq. (4.27), which can not exceed the available buffer in the electronic layers. i.e., the differential delay between any two spectrum paths used for a connection can not exceed M (Eq. (4.27)).

$$\forall p \in \mathcal{P}, e \in E : z_{p,e} \leq x_{p,e} \cdot |F| \tag{4.23}$$

$$\forall p \in \mathcal{P}, e \in E : z_{p,e} \leq T_p \tag{4.24}$$

$$\forall p \in \mathcal{P}, e \in E : z_{p,e} \geq T_p - (1 - x_{p,e}) \cdot |F| \tag{4.25}$$

$$pd_p = \sum_{e \in p} LD_e \tag{4.26}$$

$$\forall p, p' \in \mathcal{P} : |pd_p - pd_{p'}| + (GVD_p + GVD_{p'}) \leq M \tag{4.27}$$

The complexity of an ILP model is known to be exponential, i.e., $O(2^n)$, where n is the number of variables. Thus the proposed ILP model for parallel routing and spectrum assignment for parallel transmission has an exponential complexity of $O(2^n)$ with n in $O(|P| \cdot (|P| + |E| \cdot |F|))$, where $|P|$ is the number of computed paths, $|E|$ and $|F|$ are number of links and number of sub-carriers, respectively. It makes the optimization model computationally expensive and rather infeasible in practice. For example, assume a network with $|V| = 15$ and $|F| = 16$, there are $|V| \cdot |V - 1| = 210$ node pairs. For each node pair, there are $|P| \cdot |F|$ instances of $y_{p,i}$. When four paths are computed for the connection, i.e., $|P| = 4$, there are 13440 $y_{p,i}$ variables. Other variables can be calculated in a similar way. The total number of variables is very high even for small networks.

The problem size can be reduced by pruning the variables. A common method used in the literature is to compute a set of paths in advance and use them as the input to the ILP model. For instance, if 10 paths are computed in advance and 4 paths are selected as a solution from the optimization, the number of $y_{p,i}$ variables is reduced to be $C_{10}^4 \cdot |i| = 3360$. However, the solutions are limited in the pre-computed path set in this case and the complexity of the advance path computation should also be considered.

4.7.2 Heuristic Algorithm

In this section, we propose a heuristic algorithm [7] which decomposes the MRSA problem into two sub-problems, i.e., multipath

computation (Alg. 5) and spectrum assignment (Alg. 6). The objective and constraints of ILP model are also considered in the heuristic algorithm. The maximum number of fiber-level paths is limited to K for each connection request in order to reduce the complexity.

Algorithm 5: Multiple Fiber-level Path Computation

Input: $G(V, E), K,\ R(S, D)$
Output: Fiber-level Path(s) for R
Parameters: $\mathcal{S}$ as an ordered (via delay) set of paths starting from source S;
while *(*$|\mathcal{FP}| \leq K$*)* **do**
  while *destination*$(fp) \neq D$ **do**
    Select min-delay path fp from $\mathcal{S}$
    for *all nodes* v' *connected to destination*(fp) **do**
      if *(*v' *not traversed in* fp*)* **then**
        create fp' by extending p to v'
        add fp' to $\mathcal{S}$
      end
      Put fp into $\mathcal{FP}$
      Remove fp from $\mathcal{S}$
    end
  end
end
Return $\mathcal{FP}$

The proposed heuristic algorithm for multipath computation is shown in Alg. 5, which computes a set of fiber-level paths. The output of Alg. 5 is used as input to the spectrum assignment algorithm shown in Alg. 6. The algorithm starts from collecting all fiber links originating from source node S. All outgoing links from S are placed in a set denoted as $\mathcal{S}$ and sorted in an increasing order of path delay. The shortest path in $\mathcal{S}$, denoted as fp, is selected and extended to all the nodes connected to the sink node of fp, i.e., *destination*(fp). Afterwards, the path set $\mathcal{S}$ is updated with the extended links and the shortest path from current $\mathcal{S}$ is selected. The same procedure

is repeated till the shortest path in $\mathcal{S}$ reaches the destination node D. The computed path fp is placed in fiber-level path set $\mathcal{FP}$ and removed from $\mathcal{S}$.

Algorithm 6: Spectrum Assignment

Input: $G(V, E)$, $R(S, D, T_r)$ and $\mathcal{FP}$
Output: One or multiple spectrum path(s) for R

```
1  //Step 1: Single spectrum path first;
2  for k = 1 to K, fp_k ∈ FP do
3  |  Identify the spectrum path with maximum consecutive
   |  sub-carriers, i.e., p_k;
4  |  if F(p_k) ≥ T_r then
5  |  |  A single spectrum path found; break;
6  |  end
7  end
8  //Step2: Multiple Spectrum Paths
9  for k = 1 to K, fp_k ∈ FP do
10 |  for all e_i ∈ fp_k do
11 |  |  Find spectrum paths on the fiber-level path fp_k and put in
   |  |  the path set P_k
12 |  end
13 end
14 for k = 1 to K, fp_k ∈ FP do
15 |  Sort all available spectrum paths in the increasing order of
   |  delay; and put in path set P
16 |  N = |P|
17 end
18 for k = 1 to N do
19 |  if pd_{p_k} − pd_{p_1} ≤ M then
20 |  |  F+ = F_{p_k}
21 |  |  if F ≥ T_r then
22 |  |  |  Return spectrum paths and break;
23 |  |  end
24 |  end
25 end
```

The algorithm continues to select the shortest path from the up-

dated $\mathcal{S}$ and repeats the path computation. It breaks when no path can be computed or K fiber-level paths have been computed. In the worst case scenario, Alg. 5 has to visit all network nodes to find a path fp between S and D. Assume the maximum node degree in the network is $Deg(V)$, the complexity of multipath computation in Alg. 5 is $O(|V|^2 \cdot Deg(V) \cdot K)$.

The output from Alg.5, i.e., $\mathcal{FP}$, is used as input to Alg. 6 which tries to find a single spectrum path for the connection request first. It identifies the spectrum path with maximum number of consecutive sub-carriers on each fiber-level path $fp \in \mathcal{FP}$ and compares the available bandwidth with T_r. When it fails to find a single spectrum path, the algorithm continues to find a multipath solution, i.e., aggregating spectrum fragments from multiple spectrum paths. All spectrum paths in $\mathcal{FP}$ are sorted in the increasing order of delay and put in the set P. Afterwards, the differential delay and bandwidth constraints are checked. If the differential delay between a spectrum path $p_k \in \mathcal{P}$ and the shortest path $p_1 \in \mathcal{P}$ is not larger than M, i.e., $pd_{p_k} - pd_{p_1} \leq M$, p_k is included in the solution. Note that the GVD caused differential delay is not considered here. The algorithm outputs a solution when bandwidth requirement is satisfied. In the worst case scenario, Alg. 6 has to check all sub-carriers over all fiber links. Hence, the computational complexity of spectrum assignment phase is in $O(|K| \cdot |F| \cdot |E|)$.

4.7.3 Performance Evaluation

This section presents the performance evaluation of the algorithms proposed in this chapter. The connection requests arrive following a Poisson process with an average arrival rate of u and the holding time of each connection request follows the negative exponential distribution with an average value of h time units; thus the traffic load in the network is quantified as $u * h$ in $Erlang$. In our study, the mean inter-arrival time of the connection requests is 1 time unit

and mean holding time is varying to achieve different network loads. Blocking probability is used as metric for assessing the performance, which is defined as the percentage of blocked connection requests versus total connection requests.

The ILP model is implemented in Gurobi Optimizer [24] and the heuristic algorithm is evaluated with an event-driven simulator implemented in Java. We consider two representative values as the maximum acceptable differential delay, i.e., 250 μs as suggested in ITU-T G.709 [17] and 128 ms which can be supported by commercial framer mappers [44]. The evaluation uses US backbone topology [1] (Fig. 3.9). The results shown in the following section for heuristic algorithm have a confidence interval of 95%. We study the impact of different number of sub-carriers per fiber ($|F|$) and the maximum number of fiber-level paths (K) as well as maximum acceptable differential delay (M).

4.7.4 Evaluation of ILP Model

Given the complexity issue of ILP optimization, we first evaluate the proposed ILP model in a scaled-down network scenario where an optimal solution can be obtained within a reasonable time. The number of sub-carriers per fiber link is 16; the maximum differential delay is 128 ms and guard band is set to be one sub-carrier. The heuristic algorithm is also evaluated in the same experimental setting and compared with the ILP model. Afterwards, we study only the heuristic algorithm in a scenario where the ILP model becomes intractable and thus of little practical relevance. The traffic load is generated by the connection requests following Poisson process with bandwidth requirement uniformly distributed between 1 and 4 sub-carriers. When the network is stable at a certain network load, a connection demand requesting between 4 and 6 sub-carriers is sent to a randomly selected source and destination pair and an algorithm is invoked. The same experiment is repeated over 50 times to obtain

Table 4.3: Blocking probability with the ILP model and heuristic algorithm

Load ($Erlang$)	Heuristic (K= 10)	Heuristic (K= 40)	ILP $M = 128ms$
30	16.0132%	8.0723%	0.000%
35	24.823%	10.419%	6.025%
40	34.392%	18.362%	14.025%
45	40.015%	22.753%	20.902%

a mean value.

Table 4.3 shows the percentage of blocked connections at each given network load. To study the impact of pre-computed path set, the maximum number of fiber-level paths that can be used in the parallel transmission in the heuristic is set to be 10 and 40, respectively. There is no limitation of number of fiber-level paths used in the ILP evaluation, it stops when the model either finds a solution, or it is time out. It can be seen that ILP model always outperforms the heuristic algorithm when the problem is tractable. For instance, none of the connection requests is blocked using ILP model when network load is 30 *Erlang*. However, 16.0132% and 8.0723% connections are blocked using the heuristic algorithm with K =10 and $K = 40$, respectively. The reduction on the blocking probability with a larger K is thanks to more available spectrum paths. With the increase of network load, the number of blocked connections increases with both the ILP model and the heuristic algorithm. However, the performance of proposed heuristic algorithm is getting close to the ILP model when the number of pre-computed paths are sufficiently large. For instance, $K = 40$ leads to around 22.753% blocking at 45 *Erlang* while the ILP model leads to 20.902% blocking at the same network load.

This set of results shows that the main disadvantage of the proposed heuristic is the limited number of pre-computed paths. How-

Table 4.4: Simulation parameters

F: number of frequency slots per link	128
GB: number of sub-carries as guard-band	0-3
T_r: number of requested sub-carriers	5,10,15, 20, 25, 30
K: maximum number of fiber-level paths	30
PT-1: with large buffer PT-2: in line with ITU-T G.709	$M = 128ms$ $M = 250\mu s$

ever, the heuristics can always find a solution in a reasonable amount of time. At the same time, we have observed that the ILP becomes infeasible when network load is high or there are more sub-carriers per fiber link.

4.7.5 Evaluation of the Heuristic Algorithm

In this section, we investigate the performance of proposed heuristic algorithm. The simulation parameters are summarized in Table 4.4. We distinguish two transmission scenarios as follows:

- *Transmission on single spectrum path (ST)*: All sub-carriers are allocated on the shortest spectrum path.
- *Transmission on multiple spectrum paths (PT)*: Sub-carriers are not restricted to a single spectrum path. The spectrum paths can be allocated on a single fiber, or on multiple fiber links.

I) Evaluation with High Bandwidth Connection Requests

We first evaluate the performance of the proposed heuristic with extremely high bandwidth connection requests. Network load is generated by the connection requests with bandwidth requirement of 5 sub-carriers, i.e., $T_r = 5$. The connection requests are uniformly distributed on the randomly selected source and destination nodes.

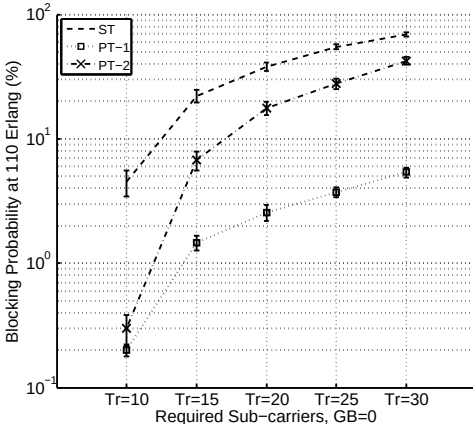


Figure 4.4: Blocking probability of high bandwidth connection requests at 110 *Erlang* with GB=0

When the network is stable at a certain network load, a connection request with high bandwidth requirement is sent to a randomly selected source and destination pair. The same experiment is repeated 1,000 times for each T_r to obtain a mean value.

Fig. 4.4 shows the blocking probability of connection requests at 110 *Erlang* with $GB = 0$. It can be seen that allowing utilization of multiple spectrum paths for a connection request can increase the acceptance of the connection requests. In case of $T_r = 10$ and only one spectrum path is allowed (ST), 4.5% connections requests are blocked. With the same setting, PT-1 and PT-2 result in 0.2% and 0.3% blocking probability, respectively, which is significantly lower. It is especially notable with larger bandwidth requirements. For instance, for connections requesting 30 sub-carriers, only 5.4% connections are blocked in the 1,000 repeated experiments with PT-1, while 69.2% connection requests are blocked with ST. With stricter differential delay constraint, PT-2 has a higher blocking probability

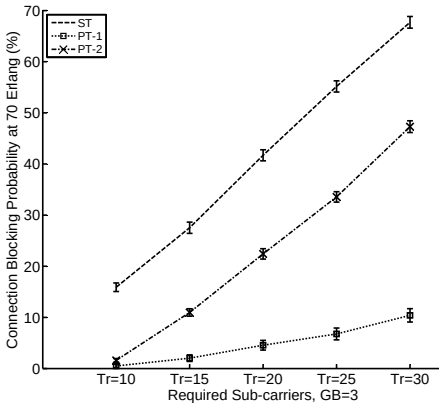


Figure 4.5: Blocking probability of high bandwidth connection requests at 70 *Erlang* with GB=3

comparing with PT-1, which is 42.3% in case of $T_r = 30$. However, it is still much lower than single spectrum path only (ST).

We use the same experiment setting to study the performance of proposed heuristic with a very large guard-band ($GB = 3$). Given the fact that the large guard-band leads to much higher blocking probability, we study the performance at a low network load. Fig. 4.5 shows the blocking probability of connection requests at 70 *Erlang* with $GB = 3$. It can be seen that the large guard-band leads to high blocking in both ST and PT, whereas PT always outperforms ST. For instance, using ST for $T_r = 10$ results in 15.9% blocking probability, while only 0.5% and 1.6% blocking probability is caused by PT-1 and PT-2, respectively. With extremely high bandwidth requirement, for instance $T_r = 30$, ST has a blocking probability of 67.7% while only 10.4% and 47.3% connection requests are blocked in case of PT-1 and PT-2, respectively.

II) Evaluation with Regular Connection Requests

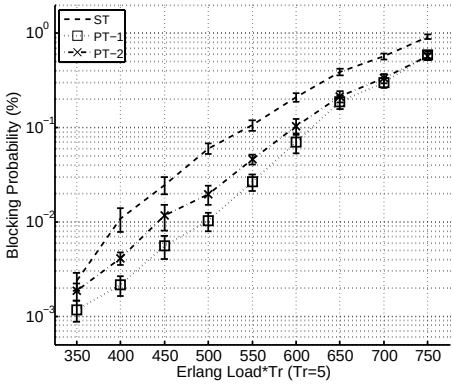


Figure 4.6: Blocking probability of regular connection requests; GB=0, Tr=5

We also evaluate the proposed heuristic algorithm in a scenario where connection requests are *regular*, i.e., don't require extremely high bandwidth. The connection requests are uniformly distributed to the randomly selected source and destination pair. For a fair comparison between connections with different bandwidth requirements (T_r), we use *relative network load* defined as $ErlangLoad * T_r$ and compare results at the same relative network loads in different settings. Average 20,000 connection requests are generated at every network load. And the same experiment is repeated 5 times at each network load.

Blocking Probability: Fig. 4.6 – Fig. 4.8 show the blocking probability of parallel transmission with different spectrum requirements. Here, we assume there is no guard-band between adjacent spectrum paths, i.e., $GB = 0$. It can be seen that using multiple spectrum paths in parallel transmission can increase the acceptance ratio of connection requests, leading to the reduction of blocking probability. With the increase of the spectrum requirement, the

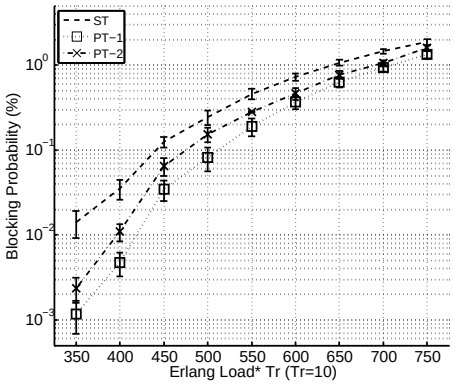


Figure 4.7: Blocking probability of regular connection requests; GB=0, Tr=10

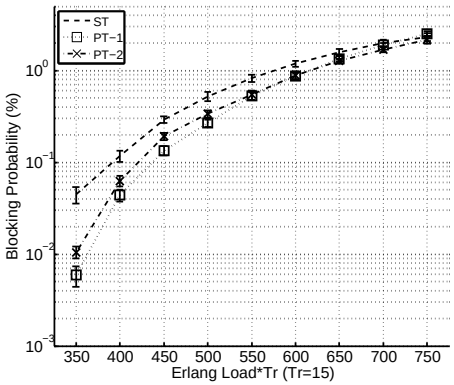


Figure 4.8: Blocking probability of regular connection requests; GB=0, Tr=15

blocking probability also increases. For instance, 1.8% connection requests are blocked with $T_r = 10$ using single spectrum path transmission (ST) at traffic load of 75 *Erlang*. With $T_r = 5$, maximum 0.9% connection requests with ST at traffic load of 150 *Erlang* as shown in Fig. 4.6. Regardless of the spectrum requirements and network loads, parallel transmission over multiple spectrum paths, i.e., PT-1 and PT-2, can always lead to a lower blocking probability, due to the fact that spectrum fragments are now aggregated instead of "wasted" in case of ST. When network load is very high, the performance of PT is also getting worse, especially when connection granularities are large. It will eventually result in the same performance as ST, as shown in Fig. 4.7 and Fig. 4.8.

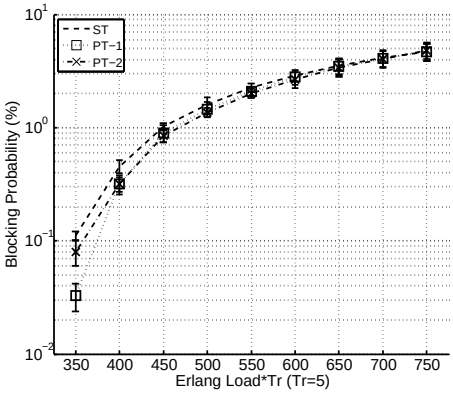


Figure 4.9: Blocking probability of regular connection requests; GB=3, Tr=5

Impact of Differential Delay Constraint: The compensation of differential delay in the upper layers can significantly affect the performance of parallel transmission in the underlying optical network. We define PT-1 and PT-2 with maximum acceptable differential delay of 128 ms and 250 μs, respectively. As it is dis-

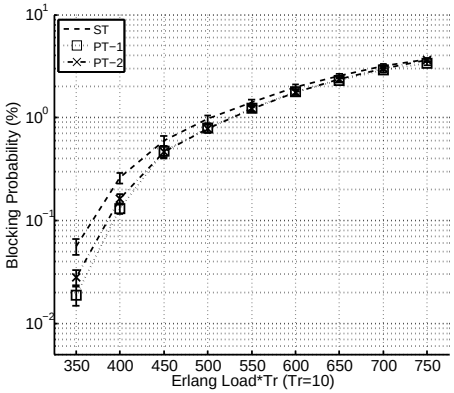


Figure 4.10: Blocking probability of regular connection requests; GB=3, Tr=10

cussed in Sec.II-A, for a connection with length of 1×10^4 km, the maximum differential delay in a standard single mode fiber is only 50 ms. Hence, the differential delay constraint of PT-1 is sufficient to utilize the long paths in the network, thus leading to the lower blocking probability, as shown in Fig. 4.6–Fig. 4.8. For example, when $T_r = 10$ and network load is 75 $Erlang$, blocking probability of PT-1 is 0.2% less comparing with PT-2, as shown in Fig. 4.7. However, since PT-1 consumes more spectrum resources by using longer paths, and it will fail to find a solution for a request requiring large number of sub-carriers when network load is high. As shown in Fig. 4.8, with $T_r = 15$, the blocking probability of PT-1 is 0.4% higher than PT-2 at network load of 50 $Erlang$.

Impact of Guard-band: Parallel transmission with multiple spectrum paths may consume more resources on guard-bands, counteracting its benefits. Fig. 4.9, Fig. 4.10 and Fig. 4.11 show the performance of parallel transmission with single and multiple spectrum paths in terms of blocking probability. It can be seen that both PT-

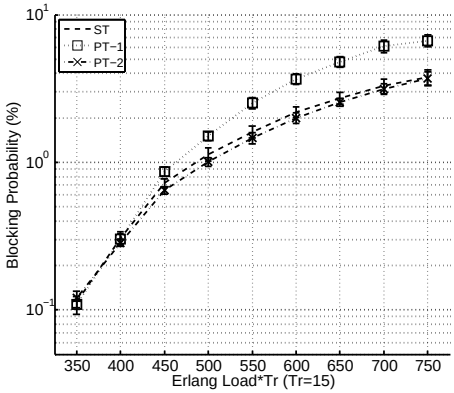


Figure 4.11: Blocking probability of regular connection requests; GB=3, Tr=15

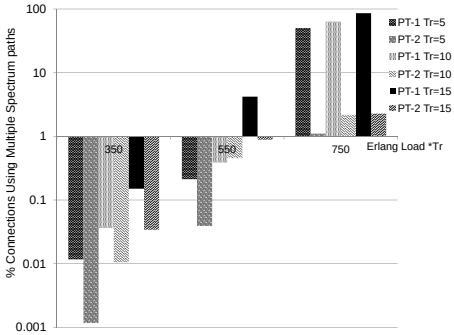


Figure 4.12: Percentage of connections served with multiple spectrum paths in parallel transmission with GB=0

1 and PT-2 still have a lower blocking probability, comparing with single spectrum path only (ST). However, with increasing network load, the resource consumed by guard-bands in PT (both PT-1 and PT-2) leads to the degradation of performance. As shown in Fig.

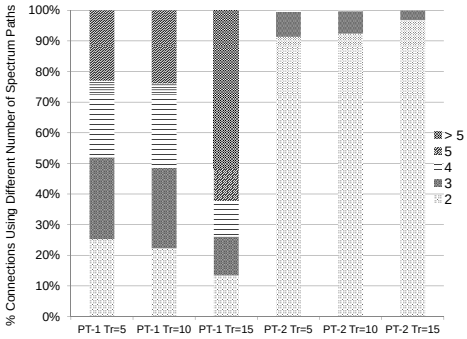


Figure 4.13: Percentage distribution of connections in parallel transmission vs. number of spectrum paths; $ErlangLoad * Tr = 750$, GB=0

4.9 and Fig. 4.10, PT-1 has the best performance at low network load. When network load is high, e.g., 75 *Erlang* and 150 *Erlang* with $T_r = 10$ and $T_r = 5$, respectively, both PT-1 and PT-2 have almost the same blocking probability as the ST.

When spectrum requirements are high, e.g., $T_r = 15$, it is not easy to find spectrum paths for both ST and PT. While PT-1 tends to reserve more resources by allowing for long paths, it has the highest blocking probability with $GB = 3$, as shown in Fig. 4.11. However, the strict differential delay constraint (250 μs) restrict PT-2 to find a single spectrum under current network condition, leading to similar performance as ST with the increasing network load.

Spectrum Fragments Aggregation Ratio: As it has been discussed before, the proposed parallel transmission algorithm tries to aggregate spectrum fragments when it can not find a single spectrum path. Hence, the fraction of connections that are served with multiple spectrum paths implies how often the spectrum fragments are aggregated.

Definition 4.1 Spectrum fragments aggregation ratio *is defined as the percentage of connection requests served with multiple spectrum paths when parallel transmission algorithms are applied. It is affected by the bandwidth requirements and differential delay constraints.*

Fig. 4.12 presents measurements under three different loads. Here, load is defined as $Erlangload * T_r$, where T_r is spectrum requirement. As it can be seen from Fig. 4.12, the strict differential delay constraint in PT-2 (250 μs) results in less connections using multiple spectrum paths. At low and medium loads ($Erlangload * T_r = 350$ and 550, respectively), most requests (> 95 %) are served with a single spectrum path in the case of PT-2, the remaining requests are served mostly (> 98 %) using only 2 spectrum paths. In case of medium load ($Erlangload * T_r = 550$) with a large spectrum requirement ($T_r = 15$), PT-1 aggregates spectrum fragments more frequently comparing with PT-2. Around 55% connection requests are served with 2 spectrum paths and 25% connections are using 3 spectrum paths. 11% connections are set up with 4 spectrum paths and the others use higher number of spectrum paths. In case of PT-1, maximum 10 spectrum paths were used for connection requests.

This behavior is even more pronounced at high network loads. The distribution of the requests served by multiple spectrum paths vs. the number of spectrum paths used is presented in Fig. 4.13. As it can be seen in Fig. 4.13, PT-2 uses only 2 spectrum paths for most connection requests, while a small fraction of connections using 3 spectrum paths. However in the case of PT-1, as the requirement for spectrum slots increases, the fraction of connections using more spectrum paths also increases, with some connections also recorded using as many as 15 spectrum paths, each with a single spectrum slot. The unbounded nature of the number and length of the spectrum paths in PT-1 imply that algorithm can provision spectrum slots across disproportionately longer paths as compared to the single path algorithm (ST). As a result, it leads to blocking

Figure 4.14: Abilene network topology [1]

of future connection requests, degrading the performance at high network loads.

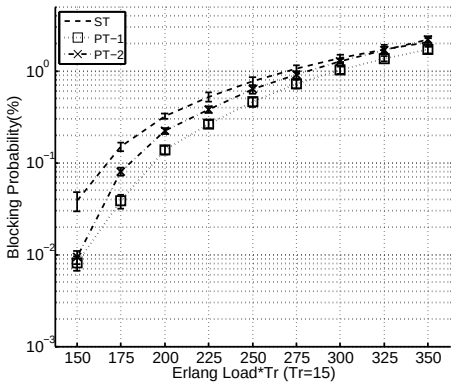


Figure 4.15: Blocking probability with GB=0 in Abilene network

Impact of Network Topologies: Finally, we evaluate the parallel transmission in a different network, namely Abilene network as shown in Fig. 4.14. Blocking probability is shown in Fig. 4.15 and Fig. 4.16, where large number of sub-carriers are required by connection requests ($T_r = 15$). It can be seen that using multiple spectrum paths can reduce blocking probability in general.

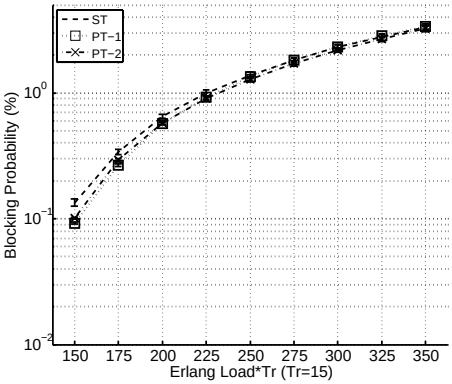


Figure 4.16: Blocking probability with GB=3 in Abilene network

In the small network, large maximum acceptable differential delay leads to better performance in terms of the blocking probability too. As shown in Fig. 4.15, PT-1 can still reduce around 0.4% blocking probability comparing with ST at high network loads (e.g., $ErlangLoad * T_r = 350$), while PT-2 has almost the same performance as ST due to the strict differential delay constraint.

4.8 Distance-Adaptive Modulation Format Assignment

This section models Multipath Routing and Spectrum Allocation (MRSA) with distance-adaptive modulation assignment [40]. By using multipath routing, we address the spectrum fragmentation issue, while making full use of OSNR margin on each spectrum path by adaptively assigning modulation format based on the path length. We propose two heuristic algorithms to study the effectiveness of multipath routing with distance-adaptive modulation format assignment. In the first algorithm, multiple paths are computed sequentially. Upon finding a path, a modulation format is assigned and spectrum is allocated for as many paths as need to fulfill the band-

width requirement. In the second algorithm, the goal is to allocate spectrum with a set of pre-computed paths with consideration of the differential delay constraint.

4.8.1 Distance-adaptive Modulation Format Assignment

Optical transmission systems are commonly designed with consideration of the worst case scenario in terms of transmission performance [6]. A typical design parameter is optical signal-to-noise (OSNR). To ensure that optical signals can be correctly received, the OSNR requirement of the longest path is considered for the whole network. As a result, an OSNR margin exists in shorter paths, which can be utilized to improve the spectrum efficiency by adaptively assigning modulation format based on channel condition. When only transmission loss is considered, the relation between distance and transmission rate can be given as:

$$C_l = C_0/2 \cdot (1 + log_2(2L/L_1)) \tag{4.28}$$

where C_0 and L is the capacity and distance of the reference path [45]. Hence, if a frequency slot of 12.5GHz, for instance, can be used to transmit data rate of 40Gbps for a path length of 500km, the same frequency slot can be used to transfer traffic at date rate of 80Gbps within the same OSNR margin when the path length is 250km. Prior work in [6] has in fact shown the validity of distance adaptive spectrum assignment.

The length of a spectrum path is the sum of length of all fiber links it traverses, which is defined as $L_p = \sum_{e \in p} L_e$. The modulation formats that can be supported by the network are denoted as m_k, $m_k \in \mathcal{M}$. For each modulation format, there exists a maximum transmission distance, denoted as L_{m_k} [6]. A modulation format m_k can be applied to a spectrum path p, only if $L_p \leq L_{m_k}$. Otherwise, a modulation format with lower modulation level has to be used. A connection demand is represented as $R(S, D, C)$, where S, D and C

are source, destination, the required transmission data rate, respectively. The number of sub-carriers that are required by a request R depends on the used modulation format. Assume the bit rate per symbol of a modulation format m_k is denoted as b_{m_k}, then the number of sub-carriers (N_{sub}) for a traffic demand can be calculated as follows:

$$N_{sub} = \lceil C/(b_{m_k} \cdot s_f) \rceil \tag{4.29}$$

where s_f is spectrum width of a frequency slot. An assumption hold in the distance-adaptive modulation assignment in this chapter is that the power injected in transponders for each channel is constant. It is due to the fact that modifying power individually for each channel can affect the gain and penalty of other channels which share the same optical amplifier in the same fiber link [46]. Therefore, the data transmission rate of each sub-carrier channel only depends on the modulation level, i.e., bits per symbol.

4.8.2 Heuristic-I: Without pre-computed paths

The first heuristic algorithm is shown in Alg.7. It starts from computing the shortest fiber-level path $\tilde{fp}$ with available continuous spectrum. It then assigns a best modulation format, e.g., m_j, depending on the length of $\tilde{fp}$. The modulation format is assigned by comparing the path length with the reference distance of available modulation formats in $\mathcal{M}$. For instance, if $Lm_{j-1} < L_{\tilde{fp}} \leq L_{m_j}$, m_j is the best option for spectrum paths on the fiber-level path $\tilde{fp}$. The required number of sub-carriers can be calculated based on the bit rate of the modulation format m_j, as shown in Eq.(4.29). If there are enough sub-carriers to support the bandwidth requirement, a single path solution is found. Otherwise, it goes on aggregating available spectrum fragments on $\tilde{fp}$.

If available resource on a single fiber-level path cannot support the connection demand, the algorithm reserves all available bandwidth on $\tilde{fp}$ and computes the next shortest path in the network. If the

Algorithm 7: Heuristic-I: Multipath Routing with Shortest-Path-First Spectrum and Modulation Assignment

Input: $G(V, E), DD, \mathcal{M}, R(S, D, C)$
Output: Spectrum path(s) $\mathcal{P}$ with assigned modulation format

```
1  reservedBw = 0; P = ∅
2  while (reservedBw ≤ C) do
3      Compute a shortest fiber-level path f̃p with available
       continuous spectrum;
4      if ( ∀p ∈ P, pd_f̃p − pd_p > DD ) then
5          break;
6      end
7      while (Spectrum available on f̃p) do
8          Compute the widest spectrum path p̃ on f̃p with x̃
           consecutive sub-carriers;
9          if No Spectrum Path Found on f̃p then
10             break;
11         end
12         //assign the best modulation format;
13         if (L_{m_{j-1}} < L_f̃p ≤ L_{m_j}) then
14             Assign m_j to path fp; m_j ∈ M
15         end
16         Compute bandwidth b̃ for the spectrum path p̃ with
           sub-carriers x̃ available in M;
17         reservedBw +=b̃;
18         Mark subcarriers in selected in p̃ as reserved;
19         Add spectrum path p̃ to P;
20     end
21 end
22 return P
```

differential delay between the new path and any spectrum paths that have been reserved is larger than the predefined value DD, the computation is terminated. The connection demand is rejected in this case. Assume the total number of frequency slots on link e is F and maximum K paths can used, the complexity of Alg.1 is in $O(|V^2| \cdot F \cdot K)$.

4.8.3 Heuristic-II: With pre-computed paths

In Alg.8, the MRSA problem is decomposed into two sub-problems, namely, 1) multipath routing and modulation assignment; and 2) spectrum assignment. The two sub-problems are addressed in two steps, as shown in Alg. 8.

In Step-I, it tries to find a set of fiber-level paths and sort the paths in the increasing order of delay. We assume that maximum K fiber-level paths can be used, and all pre-computed paths are placed in $\mathcal{FP}$, where $|\mathcal{FP}| \leq K$. The routing starts from source node S are placed in a set, namely $\mathcal{S}$, which is ordered in an increasing order of delay. $\mathcal{S}$ is initialized with all outgoing links from source node. The path with shortest delay in $\mathcal{S}$ is selected, denoted as fp. It is extended to all the nodes that connected to the sink node of fp, denoted as $destination(fp)$. The path set $\mathcal{S}$ is updated with new paths. It stops till the shortest path in $\mathcal{S}$, i.e., fp, reaches destination node D. The path fp is placed into the set $\mathcal{FP}$ which will be used as input in spectrum assignment stage. The algorithm then checks the shortest path in current $\mathcal{S}$ and repeats the same procedure. It stops when there is no more available path or K paths have been found. Finally, the algorithm compares the length of each path p_k and assigns a best modulation format m_j, denoted as $p_k(m_j)$. In the worst case scenario, the Step-I of Alg.8 has to visit all the nodes in the network to find a path fp between S and D. Assume the maximum node degree in the network is $\{Deg(V)\}$, the complexity of Step-I in Alg.8 is $O(|V^2| \cdot Deg(V) \cdot K)$.

Step-II of Alg.8 starts from searching available spectrum slots on the shortest path in $\mathcal{FP}$. At this stage, we consider bandwidth requirement constraints as defined in Eq. (4.29). All available spectrum paths on fiber-level paths in $\mathcal{FP}$ are identified and sorted in the increasing number of delay in $\mathcal{P}$. In this step, it is important to consider the differential delay issue due to the diversity of the fiber-level paths involved. It starts from the shortest path and com-

Algorithm 8: Heuristic-II: Spectrum and Modulation Assignment with Pre-computed Paths

Input: $G(V, E), K$, $R(S, D)$
Output: $\mathcal{FP}$ and $\mathcal{P}$

StepI: Multipath routing and modulation assignment
Parameters: $\mathcal{S}$ as an ordered (via delay) set of paths starting from source S;
while *($|\mathcal{FP}| \leq K$)* **do**
  while *destination*$(fp) \neq D$ **do**
    Select min-delay path fp from $\mathcal{S}$
    for *all nodes v' connected to destination*(fp) **do**
      if *(v' not traversed in fp)* **then**
        create fp' by extending fp to v'
        add fp' to $\mathcal{S}$
      end
      Put fp into $\mathcal{FP}$
      Remove fp from $\mathcal{S}$
    end
  end
end
for $k = 1$ *to* K, $fp_k \in \mathcal{FP}$ **do**
  if *($(Lm_{j-1} < L_{\tilde{fp}})$* **then**
    Assign m_j to path fp_k, i.e., $fp_k(m_j)$
  end
end
StepII: Spectrum assignment
for $k = 1$ *to* K, $fp_k \in \mathcal{FP}$ **do**
  Sort all available spectrum paths in the increasing number of delay;
  and put in path set $\mathcal{P}$
  $N = |\mathcal{P}|$
end
for $k = 1$ *to* N **do**
  if $pd_{p_{k+1}} - pd_{p_k} \leq DD$ **then**
    $F+ = F_{p_k}$
    if $F \cdot b_{m_j} \geq C$ **then**
      Return spectrum paths and break;
    end
  end
end

pare the differential delay between p_k and p_{k+1} with the maximum allowable differential delay. If it meets the differential delay constraint, the available spectrum fragments will be accumulated and the algorithm moves on to the next round. A solution is found when bandwidth requirement is satisfied. In the worst case scenario, Step-II of Alg.8 has to check all the sub-carriers over all fiber links. Hence, the computational complexity of spectrum assignment stage is $O(|K| \cdot |F| \cdot |E|)$.

4.8.4 Performance Evaluation

We evaluate the proposed algorithms in a network topology shown in Fig. 3.9 with 24 nodes and 84 links [1]. The number of sub-carriers per link is 128 and the spectrum width per frequency slot is 12.5GHz. For practical reasons, only four modulation formats are considered, i.e., on-off with 1 bit/symbol, QPSK with 2 bits/symbol and 16QAM with 4 bits/symbol as well as 64QAM with 8 bits/symbol, with the reference distance of $8000km$, $4000km$, $2000km$ and $1000km$, respectively. Note that the reference distance of each modulation format is based on assumptions. Connection demands arrive following a Poisson process and are uniformly distributed among all nodes.

We assume that connection demands are extremely high in order to evaluate multipath routing algorithms. In the context of the today's high-speed Ethernet networks this would be equivalent of any value between 100Gbps and 120Gbps. The maximum allowable differential delay is set to be $250us$. The network load A (in $Erlang$) is defined as $u * h$, where u is connection arrival rate and h is the mean connection holding time. In our study, the mean inter-arrival time is $1s$ and holding time is varying to achieve different network load. In addition, we also show the performance of single path routing algorithm (SP) based on shortest- path-first. For a fair comparison, distance-adaptive modulation assignment is also applied in the single path approach.

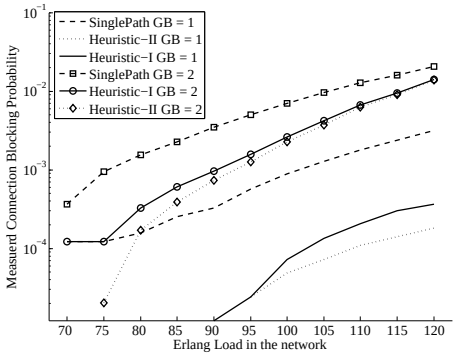
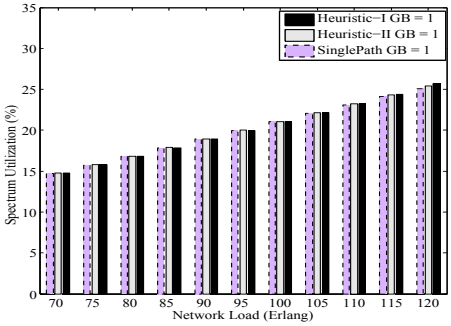


Figure 4.17: Blocking probability with GB=1,2

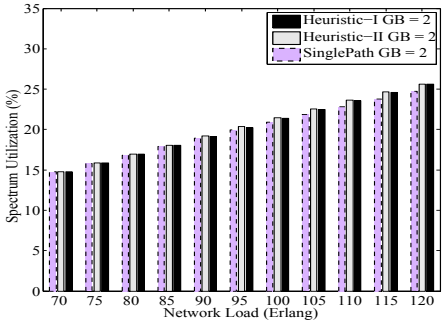
Fig. 4.17 shows the blocking probability of single path and multipath routing, with different guard-band sizes, i.e., GB=1,2. As it can be seen, the proposed multipath routing algorithms result in a lower blocking probability comparing with single path routing only. The Heuristic-II has better performance than Heuristic-I due to the fact that an optimal solution can be found. Fig. 4.17 also shows that the number of sub-carriers required as guard-band can affect the performance of all routing and spectrum assignment algorithms. A larger guard-band leads to a higher blocking probability. As shown in Fig. 4.17, the blocking probability with GB=2 is an order of magnitude larger than that of GB=1. The advantage of multipath routing is especially pronounced when network load is high. Around 1% reduction of blocking is observed with both multipath algorithms with GB=2, comparing with single path approach.

The next study focuses on spectrum efficiency. In Fig. 4.18, it can be seen that using multipath routing does not significantly increase the average spectrum usage on links, even when the guard-band size is different, which makes it highly spectrum efficient. For instance, when GB=1, both single and multipath routing approaches consume average 15% spectrum resource is used to set up spectrum

paths when network load is 70 *Erlang*, while around 25% spectrum resource is consumed by spectrum paths at 120 *Erlang*, as shown in Fig. 4.18(a). Fig. 4.17 and Fig. 4.18 show that using multipath routing can improve spectrum efficiency, since it can support more connections while consuming similar amount of spectrum resource as single path routing.



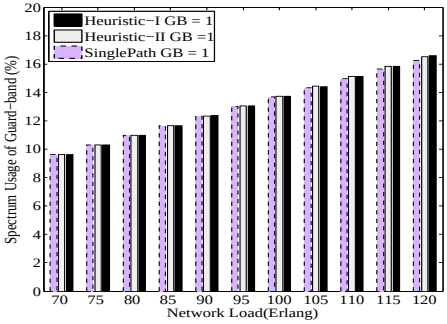
(a) GB=1



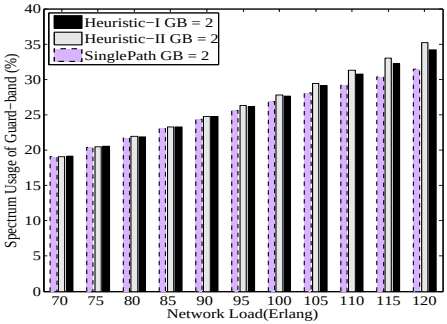
(b) GB=2

Figure 4.18: Average spectrum utilization by spectrum paths

One concern of using multipath routing in elastic optical networks is that using more paths may lead to higher spectrum consumption



(a) GB=1



(b) GB=2

Figure 4.19: Average spectrum utilization by guard-bands

on guard-band. Fig. 4.19 shows the average percentage of spectrum on link e used by guard-band, where only slight increase has been observed when network load is high and guard-band requirement is large. As shown in Fig. 4.19(b), when network load is larger that 95 *Erlang*, the proposed algorithms consume more spectrum resource on guard-band. The important phenomenon observed in this study is that guard-bands in either single path routing and multi-path routing approaches consume a substantial number of frequency

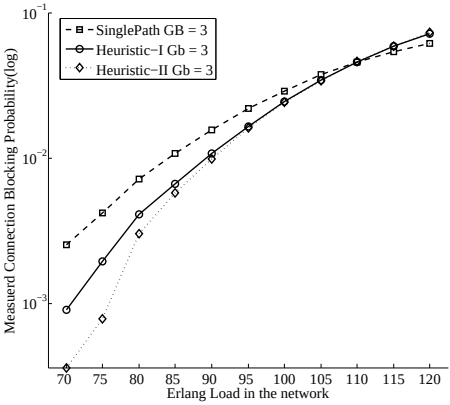


Figure 4.20: Blocking probability with GB=3

slots (Fig. 4.18(b)), which is a key issue in flexi-grid networks. To mitigate this issue, on one hand, more advanced optical network elements, such as filtering, transceivers etc, are required to reduce the required guard-band to isolate the adjacent spectrum paths. On the other hand, efficient routing and spectrum assignment algorithms are a workaround solution, such as the proposed multipath routing approaches.

Finally, it should be noted that the size of guard-band is a critical factor which should be designed with care to fully benefit from multipath routing. As shown in Fig. 4.20, using multipath can reduce the blocking probability when 3 sub-carriers are used as guard-band under certain network load, i.e., 110 *Erlang*. When network load is higher than 110 *Erlang*, multipath routing approaches result in a higher blocking comparing with single path routing due to its higher need for guard-bands. From this study, we conclude that multipath routing is more suitable for the network requiring smaller guard-band, i.e., in support of lower speeds of individual connections jointly setup in multipath routes.

4.9 Summary

This chapter investigated the technical feasibility of parallel transmission in OFDM-based optical networks to support high-speed Ethernet and designed a novel framework which is in-line with current IEEE and ITU-T standards. We formulated an optimization model based on Integer Linear Programming (ILP) for dynamic computation of multiple parallel paths and spectrum assignment, with consideration of differential delay issue caused by path diversity and fiber effects. We studied both uniform and distance-adaptive modulation assignment and proposed heuristic algorithms which can be applied for scenarios where the optimization model is intractable.

The numerical results showed that using multiple spectrum paths is especially effective in serving connection requests with extremely high bandwidth requirement, regardless of guard-band size. As the conclusion, the proposed parallel transmission framework can be used to effectively support high-speed Ethernet, while leveraging the maturity of low-speed optical technologies.

5
Parallel Transmission with Linear Network Coding

5.1 Introduction

In high-speed Ethernet, parallel transmission over multiple paths is a standardized solution in IEEE 802.3ba, enabling immediate capacity for large bandwidth flows (40/100 Gbps and more) by inverse-multiplexing into multiple lower-speed flows (e.g., 10 Gbps). From a practical perspective, parallel transmission makes networks highly backward compatible: instead of upgrading to high-speed interfaces, the existing low-speed interfaces can be fully utilized to support large connections. When multiple flows traverse different paths, attention needs to be paid to the resulting differential delay, which requires computationally expensive path optimizations and large buffering at the receiver. Frames arriving from different parallel paths need to be correctly aligned and multiplexed into a single flow at the receiver, which requires buffering. To this end, all related work has focused on optimizing the splitting ratio among paths or minimizing differential delay via optimizations, as presented in previous chapters.

In this chapter, we propose to apply linear network coding in parallel transmission, and present a novel *network coded parallel transmission* framework for high-speed Ethernet, which can lower the requirements on optimality of multipath routing, while reducing the buffer required for differential delay compensation [47]. The pro-

posed framework applies network coding in end-systems and at the price of small coding overhead, to consequently eliminate the multipath routing in the network. Extensions that are necessary to enable linear network coding in high-speed Ethernet are also proposed. Afterwards, an upper bound of buffer size required in the network coded parallel transmission is derived.

Finally, we show a case study of applying linear network coding in the parallel transmission framework proposed in Chapter 4 for optical OFDM networks. We show that optical OFDM networks can be more fault tolerant with linear network coding, in addition to reduction of buffer size. Our method can greatly increase spectral efficiency and guarantee correct reception of data packets, even in the presence of bit errors caused by optimal impairments.

5.2 Supporting Publications

1. X. Chen, A. Engelmann, A.Jukan, M.Médard, "Linear Network Coding Reduces Buffering in High-Speed Ethernet Parallel Transmission Systems," IEEE Communication Letters, Volumn 8, Issue 4, April 2014, PP. 636-639.

2. X. Chen, A. Jukan, M. Médard, "A Novel Network Coded Parallel Transmission Framework for High-Speed Ethernet," in IEEE Global Telecommunications Conference (Globecom), December 2013.

5.3 Architectural Extension to IEEE 802.3ba

To enable linear network coding in the parallel transmission, extensions are required in the high-speed Ethernet architecture specified in IEEE 802.3ba, which is shown in Fig. 5.1.

The proposed linear network coding and decoding modules are placed in the electronic layer. Linear network coding after the PCS

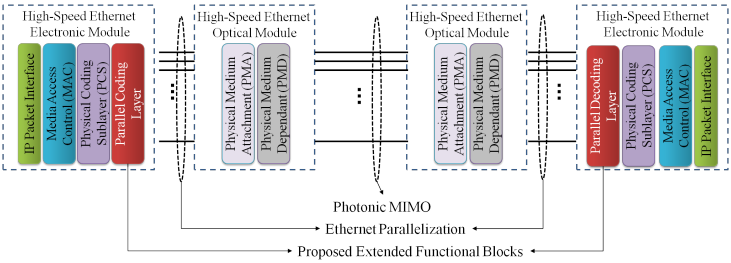


Figure 5.1: Reference architecture for high-speed Ethernet parallel transmission with linear network coding

module which encodes 66b data blocks from parallel Ethernet lanes and send them to multiple optical channels. The decoding at the destination happens before the PCS layer. After the decoding, original data blocks are sent to PCS layer which ensure the output order of the data blocks. When the linear network coding can reduce or eliminate the impact of differential delay resulted from using multiple paths in the interconnecting network, de-skew and ordering in PCS layer can be significantly simplified and accelerated.

5.4 Linear Network Coding Model

We adopt the network model from [48] which represents a network as a directed and acyclic graph $G(V, E)$. V and E are vertex set and edge set of the network, respectively. We intend G not to represent the full network, but only contains the nodes selected from the network for transmission. Hence, the assumption that G is directed and acyclic is reasonable. When e_j is an incoming link of node v, it is denoted as $head(e_j) = v$; likewise, when e_j is an outgoing link of node v, it is denoted as $tail(e_j) = v$.

The linear coding process is performed over a finite field F_{2^q}, where 2^q is the field size. Hence, traffic as a binary sequence is decomposed into *symbol* sequence with each symbol of the same length q. Sym-

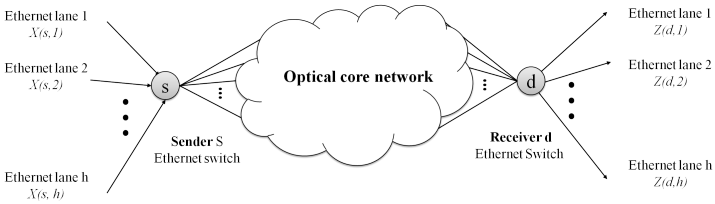


Figure 5.2: System model for network coded parallel transmission

bols on parallel lanes encoded with the same set of coding coefficients are referred to as in the same *generation*. The time unit is chosen based on the link capacity of the physical link, such that capacity of any edge in G is q bits per unit time. For instance, if the physical network is an OC-192 network with 10Gbps per wavelength, and assume finite field F_{2^8}, then the time unit is 0.8 ns. An optical link with multiple wavelengths is modeled as parallel links with unit capacity per time unit.

Our model is designed for the scenario where high-speed Ethernet parallel transmission utilizes the existing optical infrastructure, e.g., the parallel transmission is over a WDM network, as it is shown in Figure 5.2. We assume that all nodes in optical core networks are perfectly synchronized in terms of symbol timing and there is no buffering process inside the optical network, and each node simply linearly combines received symbols and sends out. Decoding only happens at the destination node where a buffer is required to ensure symbols from the same generation received upon the decoding. For an easy understanding of our model, we clarify the used notations as follows:

- ξ is the set of all links in G.
- $\mathcal{I} = \{1, 2, ..., h\}$ is set of Ethernet lanes.
- $\mathcal{A} = \{a_{i,e_j}\}$ is a $h \times \xi$ matrix contains all coefficients used at

the source node; $i \in \mathcal{I}$ and $tail(e_j) = s$.

- $\mathcal{B} = \{b_{e_j,i}\}$ is a $h \times \xi$ matrix contains all coefficients used at the destination node; $i \in \mathcal{I}$ and $head(e_j) = d$.
- $\mathcal{F} = \{f_{e_i,e_j}\}$ is a $\xi \times \xi$ matrix contains the network information. $f_{e_i,e_j} = 1$ if link e_j has input information from e_i, otherwise, $f_{e_i,e_j} = 0$.
- $X^t(s,i)$ is information at source node s at time t on lane i.
- $Y^t(e_j)$ is information on link e_j at time t.
- $Z^t(d,i)$ is the decoded information at the destination d on output Ethernet lane i at time t.
- δt is the decoding interval, i.e., the decoding process happens every δt time units.

At the source node, all the information is generated by the processes $X(s,i)$ on every virtual Ethernet lane i with an identical and constant entropy of q bits per unit time. The number of Ethernet lanes, denoted as h, is decided by the end-system. For instance, $h = 4$ in case of 40GE standard [2]. Unlike the conventional parallel transmission where data blocks are sent to each lane in round robin fashion, our model with network coding takes information generated on all Ethernet lanes in parallel and encoded with simple linear coding. The signal carried on an outgoing link of s at time $t+1$ is denoted as $Y^{t+1}(e_j)$, $s = tail(e_j)$ is represented as follows:

$$\forall e_j : tail(e_j) = s : \quad Y^{t+1}(e_j) = \sum_{i:\mathcal{I}} a_{i,e_j} \cdot X^t(s,i) \tag{5.1}$$

Given the fact that optical paths are dedicated for each connection, hence there is no information injected in the intermediate nodes. The linear coding process in an optical node can be simply performed with optical logical gates for XOR operation. In our

model, we assume the optical nodes don't wait for the information while simply perform XOR operation based on the static matrix. This operation is then similar to the optical switching function. The feasibility has been discussed and reported in [49]. The coding process at an intermediate node v at time $t+1$ is represented as follows:

$$\forall e_j : tail(e_j) = v : Y^{t+1}(e_j) = \sum_{e_i : head(e_i) = v} f_{e_i,e_j} \cdot Y^t(e_i) \qquad (5.2)$$

Given the different transmission delays of all packets of the same generation due to different link delays, decoded information at time t on the output lane i is modeled as follows:

$$Z^t(d,i) = \sum_{e_j : head(e_j) = d} \sum_{u=0}^{\delta t+1} b_{i,e_j} \cdot Y^{t-u}(e_j) \qquad (5.3)$$

Where a_{i,e_j}, f_{e_i,e_j} and b_{i,e_j} are randomly chosen from the finite field F_{2^q} and collected in the matrices $\mathcal{A}$, $\mathcal{F}$ and $\mathcal{B}$, respectively. A triple $(\mathcal{A}, \mathcal{F}, \mathcal{B})$, referred to as a *linear network code* [50], specifies the parallel transmission process between two Ethernet switches over a core network.

5.5 Replacement of Optimal Multipath Routing

Instead of optimized multipath routing, we propose to represent a network as an adjacency matrix which specifies how signals can be transmitted between the adjacent links. We label all the links in G ancestrally, i.e., the upstream links always have a lower number. For instance, if $tail(e_i) = v = head(e_j)$, then e_j is the upstream link of e_i and $j < i$. Thus, the $\xi \times \xi$ matrix $\mathcal{F}$ is a strictly upper triangular. All possible paths between source and destination can be presented as a polynomial, denoted as $\mathcal{G}$:

$$\mathcal{G} = I + \mathcal{F} + \mathcal{F}^2 + ... + \mathcal{F}^\eta \qquad (5.4)$$

Where I is an identity matrix. Let us denote $\mathcal{M}$ as the transfer matrix between the source and destination nodes, thus $\mathbf{X} \cdot \mathcal{M} = \mathbf{Z}$ and $\mathcal{M} = \mathcal{A}\mathcal{G}\mathcal{B}^T$.

Lemma 1 *For the high-speed Ethernet parallel transmission, the need for multipath routing in transmit networks can be eliminated, by designing a linear code in which the determinant of $\mathcal{M}$ is nonzero over F_{2^q}.*

Proof 5.1 *The determinant of $\mathcal{M}$, denoted as $Der\{\mathcal{M}\}$, is a polynomial in multiple variables F_{2^q} $[..., a_{i,e_j} ..., f_{e_i,e_j} ..., b_{i,e_j}]$ over the finite field F_{2^q}. Let us denote the polynomial ring over $F_{2^q}[\gamma_1, \gamma_2, ..., \gamma_n]$. For any non-zero element $f \in F_{2^q}[\Gamma_1, \Gamma_2, ..., \Gamma_n]$, there exists at least a solution $\{\gamma_1, \gamma_2, ..., \gamma_n\}$ such that the polynomial is non-zero [48]. Apply it to our case, when the linear code is constructed for the parallel transmission with $Der\{M\}$ is non-zero, there exist parameters in $\mathcal{G}$ result in a transmission solution.*

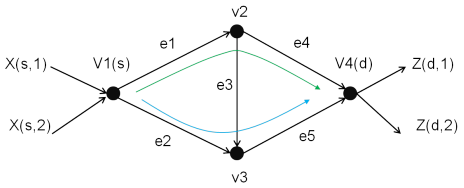


Figure 5.3: Linear network coding vs. multipath routing

An Example: We hereby illustrate this idea with a simple example based on the network shown in Figure 5.3 which shows how a linear network code is constructed for a parallel transmission session between source s and destination d. To simply the illustration, the time information is not shown here. However, the construction of linear code is the same when time information is considered. We

first label the links and construct the adjacency matrix $\mathcal{F}$ as follows:

$$\mathcal{F} = \begin{pmatrix} 0 & 0 & f_{e_1,e_3} & f_{e_1,e_4} & 0 \\ 0 & 0 & 0 & 0 & f_{e_2,e_5} \\ 0 & 0 & 0 & 0 & f_{e_3,e_5} \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

The corresponding $\mathcal{G}$ is constructed as defined in Eq. 5.4, i.e.,

$$\mathcal{G} = I+F+F^2+... = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix} + \begin{pmatrix} 0 & 0 & f_{e_1,e_3} & f_{e_1,e_4} & 0 \\ 0 & 0 & 0 & 0 & f_{e_2,e_5} \\ 0 & 0 & 0 & 0 & f_{e_3,e_5} \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix} +$$

$$\begin{pmatrix} 0 & 0 & 0 & 0 & f_{e_1,e_3}f_{e_3,e_5} \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix} = \begin{pmatrix} 1 & 0 & f_{e_1,e_3} & f_{e_1,e_4} & f_{e_1,e_3}f_{e_3,e_5} \\ 0 & 1 & 0 & 0 & f_{e_2,e_5} \\ 0 & 0 & 1 & 0 & f_{e_3,e_5} \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

At the destination d, the received data $\begin{pmatrix} Z(d,1) & Z(d,2) \end{pmatrix}$ can be represented as follows:

$$\begin{pmatrix} X(s,1) & X(s,2) \end{pmatrix} \cdot \mathcal{A} \cdot \mathcal{G} \cdot \mathcal{B}^T = \begin{pmatrix} Z(d,1) & Z(d,2) \end{pmatrix} \qquad (5.5)$$

Given two outgoing links at source node and two incoming links at the destination node in this example, we have $\mathcal{A} = \mathcal{B} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \end{pmatrix}$. The transfer matrix $\mathcal{M}$ for example shown in Fig. 5.5 is $\begin{pmatrix} f_{e_1,e_4} & f_{e_1,e_3}f_{e_3,e_5} \\ 0 & f_{e_2,e_5} \end{pmatrix}$, and the determinant of $\mathcal{M}$ is $f_{e_1,e_4} \cdot f_{e_2,e_5}$. To ensure a successful decoding, the $\mathcal{M}$ has to be invertible, i.e., $det\{\mathcal{M}\} \neq 0$, leading to $f_{e_1,e_4} = f_{e_2,e_5} = 1$. The constructed linear code results in two disjoint paths as it is shown in Fig. 5.3. More paths can be found by constructing linear code if $min\_cut\{s\_d\}$ is larger than two.

Lemma 2 *Network coded parallel transmission problem is feasible if and only of the min_cut{s_d} in G is* $\geq h$ *in the optical core network, where s, d and h are sender, receiver and number of Ethernet lanes, respectively.*

Proof 5.2 *In order to recover the native symbols from the encoded symbol at d, the transfer matrix* $\mathcal{M}$ *is required has with a rank h, i.e., at least h innovative symbols have to be received in d. If the min_cut{s_d} is less than h, then it is not possible to deliver h innovative symbols between source and destination.*

Theorem 1 *For a feasible parallel transmission problem on an acyclic network with h Ethernet lanes and a network code with coding coefficients are in the field* F_{2^q}*, the probability that multiple paths exist between s and d on which the source information can be successfully transferred to the destination is at least* $(1 - h/2^q)$*, for* $2^q > h$ *and* $h \leq \eta$*, where* η *is the capacity of min-cut between source s and destination d.*

Proof 5.3 *As it is shown in Eq.* (5.4)*, any element in the matrix* $\mathcal{G}$ *represents a path connecting source and destination. Let us denote an element in* $\mathcal{G}$ *as* g_{ij}*. Hence the determinant of* $\mathcal{M}$ *is a multivariate polynomial, denoted as* $P(M) = P(m_1, m_2, ..., m_k)$ *where* $m_k = \prod_{i:\xi, j:\xi} g_{ij}$ *with coefficients in variables* $\{a_{i,e_j}, b_{e_j,i}\}$ *over field* F_{2^q}*. Each variable in* $P(M)$ *has a degree at most h, since the transfer matrix is a* $h \times h$ *square matrix. The proof is done by induction. According to the Schwartz-Zippel lemma, the multivariate polynomial with degree of d and variables randomly selected from a finite field S, the probability of zero polynomial is* $\leq d/|S|$*. When* $k = 1$*, the polynomial* $P(M)$ *has at most h roots, hence,* $Pr\{P(m_1) = 0\} \leq h/2^q$*. Consider* $P(M)$ *as a polynomial in* m_1*, we can write* $P(M)$ *as* $P(M) = \sum_{i=0}^{h} m_1^i \cdot P_i(m_2, m_3, ..., m_k)$*. There exits some i such that* $P_i(m_2, m_3, ..., m_k)$ *is non-zero. By induction hypothesis,*

$Pr\{P_i(m_2, m_3, ..., m_k) = 0\} \leq (h-i)/2^q$. We denote $P(m_1, m_2, ..., m_k) = 0$ as event A and $P_i(m_2, m_3, ..., m_k) = 0$ as event B. When $Pr(B) \neq 0$, the degree of $Pr(A)$ is i, that is $Pr(A|B^c) \leq i/2^q$. We have $Pr(A) = Pr(A \cap B) + Pr(A \cap B^c) \leq Pr(B) + Pr(A|B^c)$. Hence, the probability of polynomial $P(m_1, m_2, ..., m_k) = 0$ is at most $h/2^q$, where 2^q is the size of the finite field size. Then the probability of successful decoding is at least $1 - h/2^q$.

5.6 Buffer Dimensioning with Linear Network Coding

5.6.1 System Model

The reference model is shown in Fig. 5.4 [47]. The source node follows the specification in IEEE 802.3ba, which scrambles Ethernet frames and groups the data into 66b data blocks. The data blocks are distributed to h Ethernet virtual lanes in a round robin fashion. A linear encoder is introduced which packetizes the data blocks, encodes and distributes the packets to N paths in parallel, $N \geq h$. The linear coding process is performed over a field F_{2^q}, where 2^q is the field size. Hence, traffic as a binary sequence is decomposed into *symbol* sequence with each symbol of the same length q. A *packet* [1] consists of N symbols and packets encoded with the same set of coding coefficients are referred to as in the same *generation*. At the destination, received packets are stored in a decoding buffer and processed by a decoder, which runs Gauss elimination over the received packets. In the decoding buffer, a Virtual Output Queue (VOQ) is generated for each generation. The decoder is a batch processor [51] which checks all VOQs for complete generations with the decoding interval δt.

The capacity of a network link is modeled as one packet (with generation ID in case of network coding) per *Timeunit* (Tu), and

[1] We adopt a generic terminology of linear network coding and each packet has the same number of symbols, which is a different concept from IP packets.

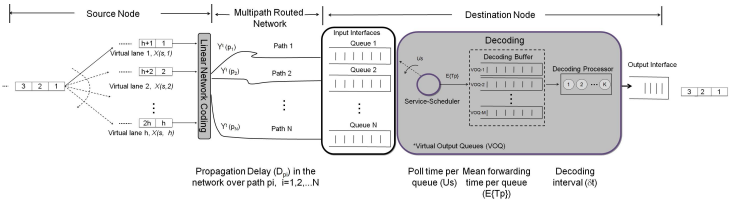


Figure 5.4: System model for network coded parallel transmission

is chosen based on the link capacity of the physical link. For instance, if the physical network is with 10 Gbps per channel, and assuming finite field F_{2^8}, then a time unit is $Tu = 0.8\ ns$ when a packet contains only a symbol. In case of a network with different link capacities, the greatest common divisor of link capacities will be chosen as the unit capacity. For instance, a link of 40 Gbps is modeled as four parallel links with unit capacity per time unit. The coding process in the network follows the definitions in Eq. (5.1), Eq. (5.2) and Eq. (5.3).

The multipath routed systems of interest are based on the circuit switching networking with parallel transmission, such as high-speed Ethernet specified in IEEE 802.3ba. In such networks, transmission rate per circuit path can be defined. As discussed earlier, the link capacity is normalized to be a packet per time unit. We hereby assume that all input queues at the destination node are $D/M/1$ queues with identical arrival rate, denoted as λ and $\lambda = 1$. Job processing follows exponential distribution. Given that the output data rate of the destination is $h \cdot \lambda$, we assume that service scheduler runs h time faster than the arrival rate of the input queue. Therefore, there are almost no packets buffered at the input queues and the size of the input buffer does not affect the decoding. In addition, we define that the service scheduler starts from the queue with corresponding shortest path at time τ_0 when the first packets arrive at the destination.

The notations used in this section are summarized as follows:

- μ_s: the mean service frequency of the service scheduler
- U_s: poll time per queue of the service scheduler
- $E\{T_p\}$: mean forwarding time of a packet
- τ_0: arrival time of the first packet at the destination
- T_{b_i}: time from τ_0 till the first packet on queue i is served
- D_{p_i}: propagation delay of path p_i
- M_{p_i}: buffer at the input interface required by p_i
- M_B: buffer required by re-ordering
- M_D: buffer required by decoding
- δt: decoding interval
- $\tilde{p}$ and p': the longest path and shortest path in the multipath routed system, respectively.

5.6.2 Upper Bound of the Decoding Buffer

The buffer in each input interface of the destination node is modeled as a First-In-First-Out (FIFO) queue. Queue i, denoted as Q_i, buffers packets from path p_i. When the system is steady, i.e., all input queues are non-empty, the cycle time of the service scheduler is:

$$T_{cycle} = 1/\mu_s = N \cdot (U_s + E\{T_p\}) \tag{5.6}$$

Buffer model before steady state: Different paths have different delays and thus exhibit different arrival time of the first packet at the destination. Before the steady state, service scheduler fetches $\sum_{i=1}^{N} x_i < N$ packets in each cycle, where x_i indicates if a packet is

fetched from Q_i in this cycle. Let us denote the shortest among the multiple paths as p_1, i.e., $p' = p_1$ and at τ_0, the first packet arrives at Q_1 and $x_1 = 1$. For $i \geq 2$, the value of x_i at time t is determined by the differential delay between p_i and p_1:

$$x_i = \begin{cases} 1, & \text{if } t \geq D_{p_i} - Dp' \\ 0, & \text{else.} \end{cases}$$

Assume the system has run m cycles at time t, i.e., $t = \sum_{n=1}^{n=m} T_{cycle}^n$. The $m+1$ cycle, denoted as T_{cycle}^{m+1} is:

$$T_{cycle}^{m+1} = N \cdot U_s + \mathcal{X}^{m+1} \cdot E\{T_p\} \tag{5.7}$$

where $\mathcal{X}^{m+1}$ is the number of packets forwarded to the decoding buffer in the cycle T_{cycle}^{m+1}; $\mathcal{X}^{m+1} = \sum_{i=1}^{N} x_i$ with

$$x_i = \begin{cases} 1, & \text{if } \sum_{n=1}^{n=m} T_{cycle}^n \geq D_{p_i} - Dp' \\ 0, & \text{else.} \end{cases} \tag{5.8}$$

The number of packets that have to be stored in the decoding buffer till it starts decoding is then:

$$M_0 = \sum_{n=1}^{K} \mathcal{X}^n \tag{5.9}$$

where $K > \left\lceil \frac{D_{\tilde{p}} - D_{p'}}{T_{cycle}} \right\rceil$, and $\mathcal{X}^n$ is determined by Eq.(5.7) and Eq.(5.8) in each cycle.

Lemma 3 *The system has to run at least K cycles, $K > \left\lceil \frac{D_{\tilde{p}} - D_{p'}}{T_{cycle}} \right\rceil$, before $Q_N \neq \emptyset$, where Q_N corresponds to the longest path in the multipath routed system.*

Proof 5.4 *In the worst case scenario, the service scheduler has to poll and process $N-1$ queues before it reaches the queue of the longest path, i.e., $\tilde{p} = p_N$. As defined earlier, the time from τ_0 till the first packet from Q_N is forwarded is $T_{b_N} = \sum_{n=1}^{K} T_{cycle}^n +$*

$N\{U_s + E\{T_p\}\}$. *According to Eq.*(5.8), $Q_N \neq \emptyset$ *if and only if* $\sum_{n=1}^{n=K} T_{cycle}^n \geq D_{\tilde{p}} - Dp'$. *And* $\forall n, n \leq K$, $T_{cycle}^n < T_{cycle}$. *Hence,* $K \cdot T_{cycle} > \sum_{n=1}^{n=K} T_{cycle}^n \geq D_{\tilde{p}} - Dp'$. *Therefore,* $K > \left\lceil \frac{D_{\tilde{p}} - D_{p'}}{T_{cycle}} \right\rceil$.

Lemma 4 *The buffer size required to start decoding is upper bounded, i.e.,* $M_0 = N \cdot (D_{\tilde{p}} - D_{p'})$, *where* $\tilde{p}$ *and* p' *are the longest and the shortest path, respectively.*

Proof 5.5 *Per Eq.*(5.8), $Q_i \neq \emptyset$ *if and only if at time* $t_i \geq D_{p_i} - D_{p'}$. *In the worst case, the decoding has to wait for the packets from the longest path, i.e.,* $\tilde{p}$ *to complete the first generation, and* $Q_N \neq \emptyset$ *if and only if at time* $t_N \geq D_{p_i} - D_{p'}$ *Hence, all packets on path* p_i *arrive during* $t_N - t_i$ *have to be stored in the decoding buffer. Denote arrival rate of* Q_i *as* λ_i, *the number of packets forwarded from* Q_i *to the decoding buffer is* $\lambda_i \cdot (D_{\tilde{p}} - D_{p_i})$. *In our model,* $\lambda_i = 1$ *packet per time unit, therefore, during* K *cycles,* $\sum_{i=1}^{N}(D_{\tilde{p}} - D_{p_i})$ *packets are stored in the decoding buffer.* $\forall p_i$, $(D_{\tilde{p}} - D_{p_i}) \leq (D_{\tilde{p}} - D_{p'})$, *the upper bound is derived.*

Buffer model during the steady state: In every decoding cycle, the number of packets arrived at the decoding buffer is:

$$M_{\delta t} = \left\lfloor \frac{\delta t}{U_s + E\{T_p\}} \right\rfloor \tag{5.10}$$

$M_{\delta t}$ is composed of packets that can complete $\gamma_{\delta t}$ generations in the decoding buffer. After each decoding cycle, $\gamma_{\delta t}$ generations are completed and released from the decoding buffer. Hence, the decoding buffer is:

$$M_D = N \cdot (D_{\tilde{p}} - D_{p'}) + M_{\delta t} - N \cdot \gamma_{\delta t} \tag{5.11}$$

The worst case scenario exists in case of linear network coding when only one generation is decoded after each decoding cycle, i.e., $\gamma_{\delta t} = 1$. In the optimal scenario, packets received at each decoding interval can complete $\gamma_{\delta t} = M_{\delta t} \cdot \frac{h}{N}$ generations, where h is the generation size, $h \leq N$.

5.6.3 Lower Bounds of Buffer without linear network coding

When all the components related to linear network coding (marked in shadow) are removed, i.e., the linear network coding block at the source node and decoding block at the destination node, the system corresponds to the conventional multipath routing. We now derive the buffer bounds for this case, and compare to the results obtained above. For a fair comparison, the packets here are the same as the packets used in a linear network coding system, i.e., packets with fixed size. Upon arrival, each packet is routed to one of the N queues accordingly, i.e., packets transmitted on path p_i is routed to queue i of interface i. A packet leaves a queue only if all packets that are sent earlier have been served. This ensures that all packets leave the system in the correct order, as they were sent out from the source node.

We consider the worst case scenario, where the path delay is in descending order of packet order. Denote the delay of path p_i as D_{p_i}, then the worst case scenario is $D_{p_1} \geq D_{p_2} \geq \cdots \geq D_{p_N}$. At τ_0, the first packet routed on the shortest path p_N arrives at the system. Packets have to be buffered on each input interface till the first packet on the longest path, i.e., p_1 is served. T_{b_1} is defined as:

$$T_{b_1} = \left\lceil \frac{D_{p_1} - D_{p_N}}{NU_s} \right\rceil \cdot NU_s + U_s + E\{T_p\}. \tag{5.12}$$

All other input queues cannot be served when the first packet on the first queue has been served. The time that the first packet on all other queues has to wait till it is served is:

$$T_{b_i} = T_{b_{i-1}} + U_s + E\{T_p\}, i = 2, 3, ..., N. \tag{5.13}$$

From Eq. (5.12) and Eq. (5.13), we can derive the time till the first packet on queue N is served as:

$$T_{b_N} = \left\lceil \frac{D_{p_1} - D_{p_N}}{NU_s} \right\rceil \cdot NU_s + N(U_s + E\{T_p\}). \tag{5.14}$$

In practice, the longest path can be connected to any input interface. Hence, the buffer required by the path p_N, i.e., the shortest path, is the minimum buffer size on each input interface to ensure the packets processed in the right order. Let us denote the size of queue N as M_{p_N}:

$$M_{p_N} = \left\{ \left\lceil \frac{D_{p_1} - D_{p_N}}{NU_s} \right\rceil \cdot NU_s + N(U_s + E\{T_p\}) \right\}. \tag{5.15}$$

Total buffer size required by the reordering M_B is the sum of all the buffers on all input interfaces, i.e., $M_B = N \cdot M_{p_N}$. To generalize the buffer model, let us denote the longest path as $\tilde{p}$ and the shortest path as p'. We can derive a lower bound of the total buffer required by a multipath routed system, i.e.,

$$M_B = N \cdot \left\{ \left\lceil \frac{D_{\tilde{p}} - D_{p'}}{NU_s} \right\rceil \cdot NU_s + T_{cycle} \right\}. \tag{5.16}$$

5.6.4 Analytical Results

Fig. 5.5 illustrates the buffer bounds required by re-ordering and by linear network coding. The maximum differential delay $D_{\tilde{p}} - D_{p'}$ is set to 125 $Timeunits(Tu)$ and the parallel transmission is set to use 10 paths; T_{cycle} is 1 tu. Fig. 5.5 shows the impact of the decoding interval δ_t on the buffer size. The size of decoding buffer, M_D (Eq. (5.11)) is normalized by the re-ordering buffer M_B (Eq. (5.16)).

Per Eq. (5.11), the size of decoding buffer depends on the number of generations (γ_{δ_t}) decoded in each decoding interval. In the worst case, the decoder can only decode one generation in each decoding interval, i.e., $\gamma_{\delta_t} = 1$.

As shown in Fig. 5.5, linear network coding reduces about 10% buffer size when $\delta_t = 0.5$ with $\gamma_{\delta_t} = 1$. However, the decoder can generally decode and release multiple generations that are complete in the decoding buffer, further reducing the size of M_D. When the

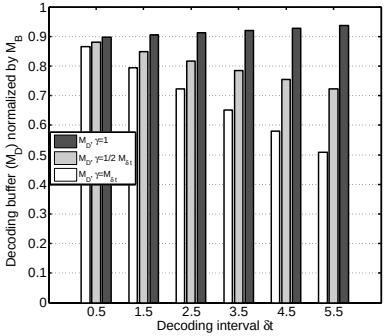


Figure 5.5: Decoding buffer M_D normalized by M_B vs. decoding interval δ_t ($T_{cycle} = 1\ Tu$, $D_{\tilde{p}} - D_{p'} = 125\ Tu$, $N = 10$)

decoder run faster than the service scheduler, i.e., $\delta_t < T_{cycle}$, the possibility of getting more than one complete generation from the decoding buffer is low. As shown in the case of $\delta_t = 0.5\ T_{cycle}$, larger γ_{δ_t} only slightly reduces the buffer. When the decoding interval is larger than T_{cycle}, more packets are stored in the decoding buffer, increasing the possibility of decoding multiple complete generations. When the decoding interval is large, on the other side, e.g., $\delta_t = 5.5\ tu$, a large number of generations are decoded and released, resulting in a small decoding buffer. The minimum size of decoding buffer is as defined in Eq. (5.9). Fig. 5.5 shows that linear network coding can reduc the buffer size without an extremely fast decoder.

5.6.5 Simulation Results

We now implement the proposed buffer model in a high-speed Ethernet system as a case study, using an event-driven simulator written in Java. The parameters used are summarized in Tab. 5.1.

In the simulation, the differential delay between a channel p_i and the shortest path is defined $i \cdot (125/4)\ \mu s$, $i = 1, 2, .., 4$. For instance,

Parameter	Value
Field size	2^8
Packet size	6 symbols with 8 bits/symbol
Generation ID	2 bytes
Number of generations	150,000
Number of parallel channels	5
Maximal differential delay	125 μs
TransmissionRate/channel	10 Gbps, i.e., time unit=6.4ns
Buffer without coding	M=1MB
Size of each input queue	10 packets

Table 5.1: Summary of parameters

assume the shortest path is denoted as p_5, then the differential delay between p_1 and p_5 is 125/4 μs. Generally, the generation size is equal to the number of paths in our model. When the generation size is smaller than the number of paths, we refer to as parallel transmission with redundancy. For instance, if the generation size is 4 and we use five paths, this extra redundancy of one more path can be used against link failures as a spare path, and also for fast decoding.

We define the *throughput* as the percentage of successfully decoded generations in all generations that are originated from the same sender node. Fig. 5.6 shows the throughput as a function of the size of decoding buffer. We can see that 100% throughput can be achieved using 60% of M_B (as calculated in Eq. (5.16)). 90% generations can be successfully decoded when the decoding buffer is only half of M_B. When the generation size is smaller than the number of paths, the required buffer is always reduced. For a larger path redundancy, for instance, for $h = 4$ and $h = 3$, 50% and buffer is reduced 60%.

Discussion and Summary: Successful decoding requires a complete generation to obtain the full rank of the coding matrix. The

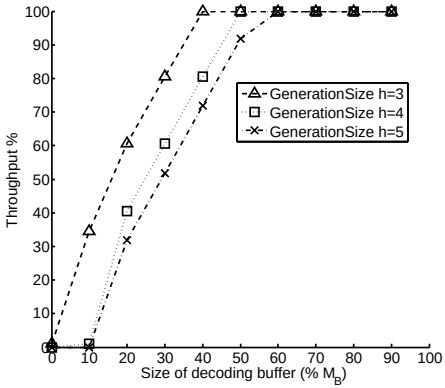


Figure 5.6: Throughput of the decoding buffer in terms of percentage of successfully decoded generations

differential delay issue in parallel transmission necessitates buffering of all packets for one generation to decode. The generation ID attached to each packet is the coding overhead. Thus, in the worst case, the number of generations that need be stored is $(D_{\tilde{p}} - D_{\tilde{p}'})/Tu$, each requiring a generation ID. As two bytes are sufficient to distinguish 65,536 generations, this overhead is a small price to pay in comparison to the saving on buffer size, especially with large packet sizes. However, the case study of high-speed Ethernet standard IEEE 802.3ba showed a reduction of 40% of the buffer size with linear network coding. With linear network coding, input interfaces of the destination node can deploy small buffers, which is critical to practical implementation of high-speed Ethernet.

5.7 Case Study: Network Coded Parallel Transmission in Optical OFDM Networks

In optical OFDM transmission systems, high-speed serial data is distributed and modulated onto multiple low-speed *sub-carriers*. For

each sub-carrier, a specific modulation format can be chosen depending on multiple factors, such as the desired spectrum efficiency, the channel condition (optical impairments) and transmission distance (optical reach). High-level modulation formats, such as 64-QAM, lead to increased bit rate per Hz (b/s/Hz, i.e., spectrum efficiency), but are less tolerant to optical impairments, such as chromatic dispersion and polarization mode dispersion. In the presence of transmission impairments that cause bit errors, signals may not be correctly detected at the receiver. This eventually leads to packet errors in the network layers, degrading the overall system performance. To correctly detect the signal at the receiver in the presence of optical impairments, spectral efficiency is traded off for optical reach [45].

We propose to apply linear network coding in parallel transmission in optical OFDM networks to increase spectral efficiency and guarantee correct reception of data packets, in the presence of bit errors caused by optical impairments. The proposed scheme uses an auxiliary parallel spectrum path to achieve fault tolerant transmission in optical OFDM networks, which as we show later, can be at much lower speeds than the main spectrum path and is thus cost-effective.

5.7.1 System Model

For the reference system model, we choose the practically relevant high-speed Ethernet traffic and OFDM optical networks for transmission (Fig. 5.7). The source node follows the specification in IEEE 802.3ba, which scrambles Ethernet frames and groups the data into 66b data blocks. The data blocks are distributed to h Ethernet lanes in a round robin fashion. Linear network coding is implemented at the sender over a finite field F_{2^q}, where 2^q is the field size.

We adopt a generic terminology of linear network coding and refer to a group of *symbols* as a *packet*, where each packet has the same size and is composed of M symbols. Furthermore, a symbol is composed of q bits. The Ethernet traffic on h lanes are encoded

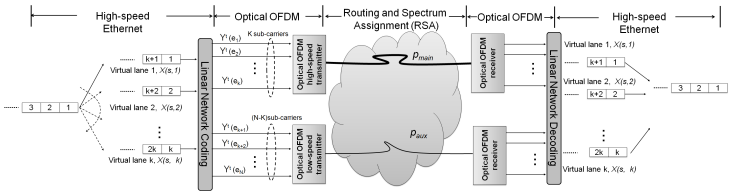


Figure 5.7: Fault tolerant transmission of high-speed Ethernet signals over optical OFDM networks with network coding; generation size $K = h$

(hence, the generation size K is equal to h) and modulated onto N sub-carriers. In this illustrative example, all sub-carriers have the same capacity (i.e, the transmission speed). However, the capacity of sub-carriers are adjustable by using different modulation formats based on the transmission distance [40]. Here, every K packets (referred to as *generation*) are encoded into N packets at the sender, where generally $N \geq K$.

Hence, the coding scheme is characterized by a *redundancy* of $N - K$ packets. As shown later, this redundancy defines the transmission reliability, i.e., the capability of the receiver to detect the correct signal in presence of optical impairments. The coding at the source node follows the process defined in Eq. (5.1) and decoding at the destination follows the process defined in Eq. (5.3). Two parallel spectrum paths are used, denoted as p_{main} and p_{aux}, with K and $N - K$ packets sent over these two paths, respectively.

The spectrum path denoted as p_{main}, is the path we would normally use to transmit the source signals without network coding and parallel transmission. Here, in addition to this spectrum path, we also use a parallel auxiliary path, p_{aux}, which is the path that can help increase spectral efficiency of the main path, i.e., p_{main}. Ideally, and we will show that this is possible, p_{aux} would need a much lower speed transceiver than p_{main} and total spectral resource is still less

comparing with normal transmission. We furthermore denote the delays and transmission speeds of paths p_{main} and p_{aux} as pd_{main}, c_{main} and pd_{aux}, c_{aux}, respectively. The ratio between the transmission speeds of these two paths, c_{main} and c_{aux}, as we will show later, determines in fact the practical applicability of our system.

5.7.2 Analysis and System Design

Thanks to the redundancy introduced by linear network coding, the proposed system is fault tolerant, i.e., it can recover any packets lost due to the transmission impairments, or due to errors, on $N - K$ packets per generation. In other words, any packet lost due to erroneous transmission on the main spectrum path, i.e., p_{main}, can be retrieved with the redundant packets received on p_{aux} from the same generation. For this concept to work, however, the redundant packets from a generation transmitted on p_{aux} should not arrive after the packets from the same generation on p_{main}, or else this would require a prohibitively large buffer at the receiver. For instance, trying to buffer packets arriving at speeds of 100GE would not be practical. That means, the spectrum path of the parallel link should exhibit lower delay or simply be chosen with a shorter optical reach. This requirement has a direct impact on the system design.

We define an important measure of fault tolerance, referred to as the *affordable packet loss (APL)*, which is the percentage of packet loss that the system can tolerate and still successfully detect all the data originally sent. The relation between random bit error and Packet Error Rate (PER) is $PER = 1-(1-BER)^{Mq}$, where Mq is the packet size. This relationship shows that the packet is considered erroneous and is dropped in the upper layers, whenever there is a bit error in the physical layer. In the following section, we will analyze the impact of our model on the optical OFDM system under given BER. If we denote the difference between the propagation delay of two spectrum paths as Δt, $\Delta t = |pd_{main} - pd_{aux}|$, the following

theorem applies.

Theorem 1 *Fault tolerant OFDM transmission system can tolerate a packet loss ratio (APL) upper bounded by* $APL \leq \frac{c_{aux}}{c_{main}} - \frac{c_{aux}\Delta t}{KMq} \leq 1$ *if* $pd_{main} \leq pd_{aux}$*; otherwise,* $APL \leq \frac{c_{aux}}{c_{main}} + \frac{c_{aux}\Delta t}{KMq} \leq 1$*, if* $pd_{main} \geq pd_{aux}$*.*

Proof 5.6 *Assume at time* t_0*,* N_{main} $(\geq K)$ *and* N_{aux} *encoded packets from one generation are sent to* p_{main} *and* p_{aux}*, respectively, where* $N = N_{main} + N_{aux}$*.* N_{main} *packets on* p_{main} *would arrive at the receiver at* $t_{main} = \frac{N_{main}Mq}{c_{main}} + pd_{main}$*, while* N_{aux} *packets on* p_{aux} *would arrive at the receiver at* $t_{aux} = \frac{N_{aux}Mq}{c_{aux}} + pd_{aux}$*, where* Mq *is the packet size. As discussed earlier, it is required that* $t_{aux} \leq t_{main}$*, i.e., the redundant packets from a generation transmitted on* p_{aux} *should not arrive after the packets from the same generation on* p_{main}*. Since the minimum value of* N_{main} *is* K*, we can hereby derive the upper bound of* $APL = \frac{N_{aux}}{N_{main}} \leq \frac{c_{aux}}{c_{main}} - \frac{c_{aux}\Delta t}{KMq}$*. The latter value becomes* $\frac{c_{aux}}{c_{main}} + \frac{c_{aux}\Delta t}{KMq}$ *when* $pd_{main} \geq pd_{aux}$*. In both scenarios, maximum affordable packet loss on* p_{main} *is 100%, i.e.,* $APL \leq 1$*.*

As it can be seen, with the larger Δt, the system can tolerate higher packet loss for the case when $pd_{main} \geq pd_{aux}$. However, per Theorem 1, APL is bounded by 1. Beyond certain point, an increase in Δt will not improve performance.

5.7.3 Analytical Results

Fig. 5.8 illustrates the affordable packet loss ratio of the system with different capacities of c_{aux}. We assume that $c_{main} = 100$ Gb/s and packet size is set to $Mq = 12,000 bits$ which is the maximal Ethernet frame size (1,500 bytes). Though we show the results for both scenarios with $pd_{main} \leq pd_{aux}$ and $pd_{main} \geq pd_{aux}$, our focus is on the latter, as explained in Section 3. For illustration purposes,

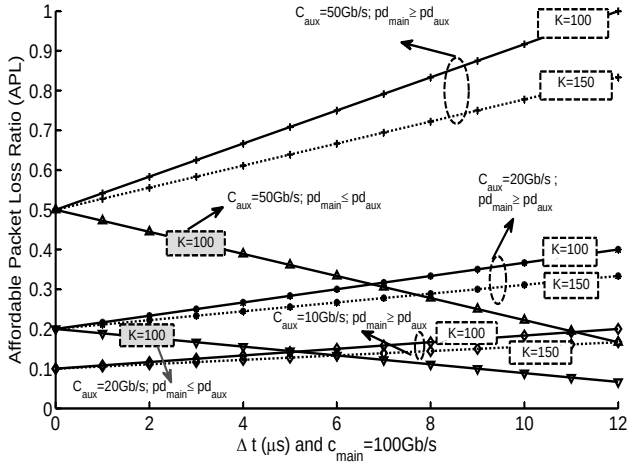


Figure 5.8: APL with $packetsize = 1500$ bytes

Fig. 5.8 only shows two cases for $pd_{main} \leq pd_{aux}$ with $c_{aux} = 50$ Gb/s and 20 Gb/s, respectively. It can be seen that a spectrum path with transmission rate of 50 Gb/s is tolerant to a maximum of 50% APL. However, per Theorem 1, the increase of differential delay will reduce the packet loss tolerance when $pd_{main} \leq pd_{aux}$. When the differential delay is large enough, e.g., $12\mu s$, the system shows very insignificant advantage comparing with single spectrum path transmission, which has very low packet loss tolerance.

When the system is designed with $pd_{main} \geq pd_{aux}$, it can, however, tolerate more packet loss with same capacity of the parallel transmission link. As shown in Fig. 5.8, a spectrum path at 10 Gb/s can ensure fault tolerant transmission of p_{main} at speed of 100Gb/s with a minimum affordable packet loss ratio of 10.08% with $K = 100$. With the increase of Δt, the tolerance to packet loss increases. For instance, when $\Delta t = 10\mu s$, APL can be up to 18.33%. When the $c_{aux} = 50$ Gb/s (half of the main capacity), the proposed system can

survive 95.83% packet loss with $\Delta t = 10\mu s$. However, the system performance reaches the optimum at $\Delta t = 12\mu s$. Further increase of Δt will not lead to any improvement, but would merely require larger buffer. We also analyze the impact of generation size on the proposed system in case of $pd_{main} \geq pd_{aux}$. With larger generation size, here $K = 150$, the system can tolerate less packet loss comparing with $K = 100$, due to the fact that the decoder needs to store and process more data for decoding.

To illustrate the improvement of spectral efficiency, Table 5.2 shows the relation between the BER and the corresponding highest modulation format can be used on p_{main} with given OSNR budget according to the data measured in [52]. The size of a sub-carrier is 12.5GHz. It can been seen that p_{aux} requires a much smaller OSNR due to the fact that it can use a low-level modulation format. The spectral efficiency on the p_{main} can be maximized by using the highest-level modulation format, while the proposed network coded parallel transmission can guarantee the correct reception at the end. For instance, when BER= 10^{-5}, p_{main} can use 32-QAM modulation format by allocating 25 GHz (2 sub-carriers).

In networks without using network coded parallel transmission, QPSK is commonly used. Instead of 2 sub-carriers, 4 sub-carriers are required. In our case, p_{aux} is modulated with BPSK using 1 sub-carrier. The total used spectral resource is 3 sub-carriers, i.e., still less than 4 sub-carriers.

BER	APL	p_{main} with OSNR=25db	p_{aux} with OSNR=15db
10^{-5}	0.113	32-QAM, 2 sub-carriers	BPSK, 1 sub-carrier
10^{-4}	0.6988	64-QAM, 2 sub-carriers	QPSK, 1 sub-carrier
10^{-3}	0.999993	64-QAM, 2 sub-carriers	16-QAM, 1 sub-carrier

Table 5.2: Modulation formats vs. BER; $c_{main} = 100$ Gb/s, $Mq = 1500\ bytes$; 12.5 GHz/sub-carrier; $pd_{main} \geq pd_{aux}$

5.8 Summary

This chapter presented a novel *network coded parallel transmission* framework for high-speed Ethernet, which can lower the requirements on optimality of multipath routing, while reducing the buffer required for differential delay compensation [47]. We showed that using linear network coding can consequently eliminate the multipath routing in the network. We proposed necessary extensions to enable linear network coding in high-speed Ethernet. We showed analytically that linear network coding can significantly reduce the buffer required in parallel transmission.

The case study of optical OFDM networks showed that by implementing linear network coding and low-speed parallel parallel channel, a fault tolerant transmission can be achieved in optical OFDM networks with increased spectral efficiency and higher tolerance to bit errors due to optical impairments. We showed that our system is fault tolerant for a spectrum path at 100Gb/s using high-level modulation formats, which makes it spectrally efficient at a small price of a low-speed auxiliary path only, allocating one sub-carrier.

6

Conclusion

The unprecedented growth in Internet traffic is driving the steep upscaling of network capacity, which even in fiber optic networks is expected to reach Shannon limit in the near future. Parallel transmission combined with multipath routing has been identified as a key solution to address the imminent capacity crunch in networks and harvest the power of end-system hardware capabilities. To this end, the presented thesis tackled the challenges and issues associated with network capacity upscaling by unifying the research on multipath routing and parallel network system design.

The thesis modeled a new multipath routing problem in optical networks for data-intensive applications, and addressed the challenges pertaining to practical implementation from all perspectives, including design of new algorithms, development of strategies for path selection and traffic splitting, as well as protocol extensions. We proposed two novel multi-domain multipath computation schemes, i.e., *segmental multipath computation* and *end-to-end multipath computation* and showed that a scalable inter-domain service provisioning in optical networks can be achieved by using multipath routing. This thesis also presented the protocol extensions and implementation of multipath routing in the standard PCE as specified by the IETF standard bodies.

The thesis presented pioneering studies on high-speed Ethernet parallel transmission in combination with various optical network technologies, from conventional WDM networks to advanced optical OFDM networks. Novel architectures were proposed, considering all the relevant design parameters including data mapping between Ethernet and optical layer, buffer availability and differential delay as well as modulation formats. Specifically, this thesis presented the first validation of compatibility between the high-speed Ethernet standard and optical OFDM networks without resource allocation penalty, such as spectrum fragmentation. We combined multipath routing in optical networks and parallelization in end-systems and modeled resource allocation and management problems for WDM and optical OFDM networks, respectively. Novel algorithms and solutions were proposed for the same.

Finally, this thesis addressed the critical issue of buffer dimensioning in high-speed parallel network systems. We designed a novel parallel transmission framework for high-speed Ethernet by applying network coding in end-systems. We analyzed the coding overhead and presented a novel buffer dimensioning. We derived an upper bound of the decoding buffer which is smaller than the buffer size required by re-ordering in conventional multipath-routed systems without linear network coding. The proposed solutions were shown to hold the key to an efficient high-speed Ethernet system design as they can significantly lower the buffer requirement at the receiver.

The future research avenues fall in the areas of designing parallel transmission frameworks with broader range of applications such as optical data centers, home networking and cyber physical systems, as well as validating the applicability of parallel transmission in networks with new technologies, such as high-speed (Gbps) wireless systems and free space optics.

Bibliography

[1] "SNDlib." [Online]. Available: http://sndlib.zib.de/home.action

[2] *Carrier Sense Multiple Access with Collision Detection(CSMA/CD) Access Method and Physical Layer Specifications*, IEEE Std. 802.3ba, 2010.

[3] X. Chen and A. Jukan, "Optimized Parallel Transmission in OTN/WDM Networks to Support High-speed Ethernet with Multiple Lane Distribution," *IEEE/OSA Journal of Optical Communications and Networking*, vol. 4, no. 3, pp. 248–258, Mar. 2012.

[4] W. D. Zhong and R.Tucker, "Wavelength Routing-based Photonic Packet Buffer and Their Applications in Photonic Packet Switching Systems," *IEEE/OSA Journal of Lightwave Technology*, vol. 16, no. 10, pp. 1737–1745, 1998.

[5] X. Chen, A. Jukan, and A. Gumaste, "On the Usage of FDLs in Optical Parallel Transmission to Support High-Speed Ethernet," in *16th International Conference on Optical Network Design and Modeling (ONDM)*, 2012, pp. 1–6.

[6] M. Jinno, B. Kozicki, H. Takara, A. Watanabe, Y. Sone, T. Tanaka, and A. Hirano, "Distance-adaptive Spectrum Resource Allocation in Spectrum-sliced Elastic Optical Path Network," *IEEE Communications Magazine*, vol. 48, no. 8, pp. 138–145, Aug. 2010.

[7] X. Chen, A. Jukan, and A. Gumaste, "Optimized Parallel Transmission in Elastic Optical Networks to Support High-Speed Ethernet," *IEEE/OSA Journal of Lightwave Technology (JLT)*, vol. 32, no. 2, pp. 228–238, 2014.

[8] P. Winzer, "Optical Networking Beyond WDM," *IEEE Photonics Journal*, vol. 4, no. 2, pp. 647–651, 2012.

[9] C. Hopps, "Analysis of an Equal-Cost Multi-Path Algorithm," IETF RFC 2992, Nov. 2000. [Online]. Available: http://tools.ietf.org/html/rfc2992

[10] A. Gumaste and N. Krishnaswamy, "Proliferation of the Optical Transport Network: A Use Case Based Study," *IEEE Communications Magazine*, vol. 48, no. 9, pp. 54–61, Sept. 2010.

[11] A. Farrel, J. Vasseur, and J. Ash, "A Path Computation Element-based Architecture," IETF RFC 4655, Aug. 2006. [Online]. Available: http://tools.ietf.org/html/rfc4655

[12] J. Vasseur and J. L. Le Roux, "Path Computation Element Communication Protocol," IETF RFC 5440, Mar. 2009. [Online]. Available: http://tools.ietf.org/html/rfc5440

[13] A. Farrel, A. Satyanarayana, A. Iwata, N. Fujita, and G. Ash, "Crankback Signaling Extensions for MPLS and GMPLS RSVP-TE," IETF RFC 4920, Jul. 2007. [Online]. Available: http://tools.ietf.org/html/rfc4920

[14] J. Vasseur, R. Zhang, N. Bitar, and J. L. Roux, "A Backward Recursive PCE-based Computation (BRPC) Procedure To Compute Shortest Constrained Inter-domain Traffic Engineering Label Switched Paths," IETF RFC 5441, Apr. 2009. [Online]. Available: https://tools.ietf.org/html/rfc5441

[15] M. Chamania, M. Drogon, and A. Jukan, "An Open-source Path Computation Element (PCE) Emulator: Design, Implementation, and Performance," *IEEE Journal of Lightwave Technology*, vol. 30, no. 4, pp. 414–426, 2012.

[16] X. Chen, Y. Zhong, and A. Jukan, "Multipath Routing in Path Computation Element (PCE): Protocol Extensions and Implementation," in *18th European Conference on Network and Optical Communications (NOC)*, July 2013.

[17] "ITU-T Recommendation G.709: Interfaces for the Optical Transport Network (OTN) ," ITU-T Recommendations.

[18] A. Jukan and J. Mambretti, "Evolution of Optical Networking Toward Rich Digital Media Services," *Proceedings of the IEEE*, vol. 100, no. 4, pp. 855–871, Apr. 2012.

[19] X. Chen, M. Chamania, A. Jukan, A. Drummond, and N. da Fonseca, "QoS-Constrained Multi-Path Routing for High-End Network Applications," in *IEEE INFOCOM Workshop (High-speed Networks)*, Apr. 2009.

[20] ——, "On the Benefits of Multipath Routing for Distributed Data-Intensive Applications with High Bandwidth Requirements and Multidomain Reach," in *Communication Networks and Services Research Conference(CNSR)*, May 2009, pp. 110–117.

[21] "ITU-T Recommendations G.707, Network Node Interface for the Synchronous Digital Hierarchy (SDH)," ITU-T Recommendations.

[22] "ITU-T Recommendations G.783, Characteristics of Synchronous Digital Hierarchy (SDH) Equipment Functional Blocks ," ITU-T Recommendations.

[23] A. Srivastava, S. Acharya, M. Alicherry, B. Gupta, and P. Risbood, "Differential Delay Aware Routing for Ethernet over SONET/SDH," in *IEEE International Conference on Computer Communications (INFOCOM)*, Mar. 2005, pp. 1117–1127.

[24] "Gurobi Optimization." [Online]. Available: http://www.gurobi.com/

[25] M. MacGregor and W. Grover, "Optimized k-shortest-paths Algorithm for Facility Restoration," *Software: Practice and Experience*, vol. 24, no. 9, pp. 823–834, 1994.

[26] S. Uludag, K.-S. Lui, K. Nahrstedt, and G. Brewster, "Analysis of Topology Aggregation Techniques for QoS Routing," *ACM Computing Surveys*, vol. 39, no. 3, 2007.

[27] "ITU-T Recommendation G.7715/Y.1706 (2002), Architecture and Requirements for Routing in the Automatically Switched Optical Network," ITU-T Recommendations.

[28] S. Sanchez-Lopez, X. Masip-Bruin, E. Marin-Tordera, J. Sole-Pareta, and J. Domingo-Pascual, "A Hierarchical Routing Approach for GMPLS Based Control Plane for ASON," in *IEEE International Conference on Communications (ICC)*, 2005, pp. 1683–1687.

[29] "Teragrid." [Online]. Available: https://www.teragrid.org/

[30] X. Chen, A. C. Drummond, A. Jukan, and N. L. da Fonseca, "Multipath Routing with Topology Aggregation for Scalable Inter-domain Service Provisioning in Optical Networks," *Optical Switching and Networking*, vol. 9, no. 4, pp. 314 – 322, 2012.

[31] J. Y. Yen, " Finding the K Shortest Loopless Paths in a Network," *Management Science*, vol. 17, no. 11, pp. 712–716, July 1971.

[32] S. Mao, S. S. Panwar, and Y. T. Hou, "On Minimizing End-to-end Delay with Optimal Traffic Partitioning," *IEEE Transactions on Vehicular Technology*, vol. 55, no. 2, pp. 681–690, 2006.

[33] A. Gumaste, "Towards a Transponder for Serial 100Gigabit Ethernet Using a Novel Optical SERDES," in *Conference on Optical Fiber Communication (OFC)*, Mar. 2009.

[34] S. Ahuja, T. Korkmaz, and M. Krunz, "Minimizing the Differential Delay for Virtually Concatenated Ethernet over SONET Systems," in *13th International Conference on Computer Communications and Networks*, Oct. 2004, pp. 205–210.

[35] M. Lukasiewycz, M. Glaß, C. Haubelt, and J. Teich, "SAT-decoding in Evolutionary Algorithms for Discrete Constrained Optimization Problems," in *IEEE Congress on Evolutionary Computation (CEC)*, 2007, pp. 935–942.

[36] K. Deb and R.B.Agrawal, "Simulated Binary Crossover for Continuous Search Space," *Complex Systems*, vol. 9, no. 2, pp. 115–148, 1995.

[37] Opt4J: Meta-heuristic Optimization Framework for Java. [Online]. Available: http://www.opt4j.org

[38] E. Zitzler, D. Brockhoff, and L.Thiele, "The Hypervolume Indicator Revisited: On the Design of Pareto-compliant Indicators via Weighted Integration," *Evolutionary Multi-Criterion Optimization, Lecture Notes in Computer Science*, pp. 862–876, 2007.

[39] M. Jinno, H. Takara, B. Kozicki, Y. Tsukishima, Y. Sone, and S. Matsuoka, "Spectrum-efficient and Scalable Elastic Optical Path Network: Architecture, Benefits, and Enabling Technologies," *IEEE Communications Magazine*, vol. 47, pp. 66–73, Nov. 2009.

[40] X. Chen, Y. Zhong, and A. Jukan, "Multipath Routing in Elastic Optical Networks with Distance-adaptive Modulation Formats," in *IEEE International Conference on Communications (ICC)*, June 2013.

[41] X. Chen, A. Jukan, and A. Gumaste, "Multipath Defragmentation: Achieving Better Spectral Efficiency in Elastic Optical Path Networks," in *IEEE International Conference on Computer Communications (INFOCOM)*, Apr. 2013, pp. 390–394.

[42] W. Shieh and I. Djordjevic, *OFDM for Optical Communications*. Academic Press, 2009.

[43] Y. Sun, T. Ono, A. Takada, and M. Tomizawa, "Wavelength-Group-Based Optical Virtual Concatenation Technique for Data-Intensive and Latency-Sensitive Applications," in *IEEE International Conference on Communications Workshops*, 2008, pp. 179–183.

[44] "Cisco SL-series." [Online]. Available: http://www.cisco.com/-en/US/products/hw/modules/ps2710/ps5479/index.html

[45] M. Kiese and M. Schuster, "Exploiting Transponder Performance in Optical OFDM Networks," in *Conference on Optical Fiber Communication (OFC)*, Mar. 2009, pp. 1 –3.

[46] Q. Yang, W. Shieh, and Y. Ma, "Bit and Power Loading for Coherent Optical OFDM," *IEEE Photonics Technology Letters*, vol. 20, no. 15, pp. 1305–1307, Aug. 2008.

[47] X. Chen, A. Engelmann, A. Jukan, and M. Medard, "Linear Network Coding Reduces Buffering in High-Speed Ethernet Parallel Transmission Systems," *IEEE Communications Letters*, no. 99, pp. 1–4, 2014.

[48] R. Koetter and M. Medard, "An Algebraic Approach to Network Coding," *IEEE/ACM Transactions on Networking*, vol. 11, no. 5, pp. 782–795, Oct. 2003.

[49] E. D. Manley, J. S. Deogun, L. Xu, and D. R. Alexander, "All-Optical Network Coding," *IEEE/OSA Journal of Optical Communications and Networking*, vol. 2, no. 4, pp. 175–191, Apr. 2010.

[50] T. Ho, R. Koetter, M. Medard, D. Karger, and M. Effros, "The Benefits of Coding Over Routing in a Randomized Setting," in *IEEE International Symposium on Information Theory*, 2003.

[51] H. Gold and P. Tran-Gia, "Performance Analysis of a Batch Service Queue Arising Out of Manufacturing System Modelling," *Queueing Systems*, vol. 14, no. 3-4, pp. 413–426, 1993.

[52] E. Vanin, "Performance Evaluation of Intensity Modulated Optical OFDM System with Digital Baseband Distortion," *Optical Express*, vol. 19, no. 5, pp. 4280–4293, Feb. 2011.

List of Symbols

$G(V, E)$	Graph of the network
V	Set of nodes in the network
E	Set of links in the network
$\mathcal{W}$	Number of wavelengths per fiber link
W_e	Set of available wavelengths on link e
LD_e	Delay of link e
pd_p	Delay of path p
$\mathcal{P}$	Set of optical paths
fp	A fiber-level path, a sequence of fiber links between source and destination
x_p	Binary variable that denotes if a path $p \in P$ is found for the connection demand
$x_{p,w}$	Binary variable that denotes if a wavelength $w \in W$ is used by the path $p \in P$
$x_{p,e}$	Binary variable that denotes if the edge e is used by path $p \in P$
$o_{p,p'}$	Binary variable that denotes if two paths $p \in P$ and $p' \in P$ share at least a link
$pd_{p,w}$	Integer variable that denotes the delay of the path $p \in P$ using wavelength $w \in W$
md	Integer variable that denotes the maximal delay in current solution

$x_{p,e,w}$	Binary variable that denotes if a path p uses wavelength w on link e
M_r	Required buffer to compensate differential delay
M_D	Available buffer at the destination node
n_{p,w,v,d_i}	Fiber delay line d_i at node v used by wavelength w in path p
s_f	Size of a frequency slot in flexi-grid optical networks
GB	Guard-band (GB)
f_i	A sub-carrier with index i
M	Maximum acceptable differential delay
$\mathcal{M}$	Set of available modulation formats
m_k	Modulation format
L_{m_k}	Maximum transmission distance of format m_k
b_{m_k}	Bit rate per symbol of a modulation format m_k
$D(f_c)$	Fiber dispersion at the central frequency
T_r	Required number of sub-carriers
$y_{p,i}$	Binary variable denoting if a spectrum path uses sub-carrier with index i (f_i)
$x_{p,e,i}$	Binary variable denoting if a spectrum path use sub-carrier with index i (f_i) on link e
GVD_p	Integer variable denoting the differential delay caused by GVD on the spectrum path p
T_p	Integer variable denoting the number of sub-carriers allocated to the spectrum path p
$head(e_j)$	e_j is an incoming link of a network node
$tail(e_j)$	e_j is an outgoing link of a network node

F_{2^q}	Finite field with field size of 2^q
ξ	Total number of links in a network

Acronyms

DWDM	Dense Wavelength Division Multiplexing
WDM	Wavelength Division Multiplexing
MLD	Multiple Lane Distribution
OTN	Optical Transport Network
ODU	Optical channel Data Unit
OVC	Optical Virtual Concatenation
VCAT	Virtual Concatenation
SONET	Synchronous Optical Networking
SDH	Synchronous Digital Hierarchy
ILP	Integer Linear Programming
FDL	Fiber Delay Line
OFDM	Orthogonal Frequency Division Multiplexing
MIMO	Multi-input-Multi-Output
BER	Bit Error Rate
SDN	Software Defined Networking
LAN	Local Area Network
OIF	Optical Internetworking Forum
QPSK	Quadrature Phase Shift Keying
MRWA	Multipath Routing and Wavelength Assignment
RWA	Routing and Wavelength Assignment
MRSA	Multipath Routing and Spectrum Allocation

RSA	Routing and Spectrum Allocation
EA	Evolutionary Algorithm
PCE	Path Computation Element
PBB-TE	Provider Backbone Bridge Traffic Engineering
ASON	Automatically Switched Optical Network
VOQ	Virtual Output Queue
QoS	Quality of Service
TED	Traffic Engineering Database
PCC	Path Computation Client
PCEP	PCE communication protocol
ERO	Explicit Route Object
ECMP	Equal Cost Multi-Path routing
PCS	Physical Coding Sublayer
AWG	Arrayed-Waveguide Grating